UDC 004

Information, Computing and Intelligent systems. – 2022. – № 3. – P. 112.

Recommended for publication by the Academic Council of the Faculty of Computer Science and Computer Engineering

The scientific journal “Information, Computing and Intelligent systems” is intended for the publication of the results of scientific research and scientific and practical developments in the field of technical sciences by students, masters, PhDstudents, scientists, and practicing specialists in the field of science "Information systems".

The thematic orientation of the journal “Information, computing and intelligent systems” is reflected in the following headings: computerized and computer systems and networks, information technologies, the Internet of Things, information transformation and processing, cloud computing, computer cryptography, data protection, intelligent systems, artificial intelligence, machine learning, automated design of software and technical tools, system control, diagnostics and control of parameters of complex systems, processes and environments; engineering knowledge, embedded systems, robotics, microelectronics.

ISSN 2708-4930

Certificate of state registration No. 23827-13667ПР from 20.02.2019
Magazine in English
Web- resource – https://itvisnyk.kpi.ua/
Format 60×84 1/8. Garnitura Times. Offset Folder № 1.

Ministry of Education and Science of Ukraine
National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”

SCIENTIFIC EDITION

Information, Computing and Intelligent systems

The journal is the legal successor of the Collection of scientific works “Bulletin of NTUU “KPI”. Informatics, control and computer engineering”

Founded in 1964 years

Issue 3

Kyiv – 2022 (3)

SUMMARY

UDC 004.383 DOI: https://doi.org/10.20535/2708-4930.3.2022.267302

DESIGN OF DATA BUFFERS IN FIELD PROGRAMMABLR GATE ARRAYS

A. Sergiyenko, P. Serhiienko, I. Mozghovyi,
A. Molchanova

The design of the data buffers for the field programable gate array (FPGA) projects is considered. A new method of buffer design is proposed, which is based on the representation of the synchronous dataflow graph in the three-dimensional space, optimization of them, and description in VHDL. The method gives the optimized buffers which are based either on RAM or on the register pipeline. The derived pipeline buffer can be mapped into the shift register primitive of FPGA. The method is built in the experimental SDFCAD framework intended for the pipelined datapath synthesis.

***Keywords**: FPGA, VHDL, synchronous dataflow, datapath synthesis.*

Introduction

Field programmable gate arrays (FPGAs) are popular devices that provide both high-speed computations for any complex task and availability for many designers of application-specific computers. The FPGA design technology was expanded over the last decades, which is based on the register transfer level (RTL) description of the computational datapath using the hardware description language like Verilog or VHDL. In recent years, high-level synthesis tools become popular because they provide a compilation of the C programs into the hardware descriptions, inviting the firmware programmers for designing the FPGA applications [1].

In many cases, the FPGA project consists of a set of ready blocks and intellectual property cores (IP cores) that communicate with each other through the proper interfaces and data buffers. But the selection of these interfaces and designing these buffers are still uncertain. In most cases, the usual methods of the RTL design are used or the ready IP cores are selected to build the data buffers, which gives the increased hardware volume, insufficient throughput, or both.

Most FPGA projects are pipelined, application-specific processors. The FPGA architecture contains a lot of hardware resources like registers, FIFOs, pipelined DSP blocks, two-port blocked RAMs (BRAM), and pipelined input-output pads, which support the pipelined computations. But they are utilized in the data buffers using the old synthesis methods which don`t provide good results. In particular, the buffers are usually designed separately from the pipelined datapath to which they are connected [2].

In the article, a new method of data buffer design is proposed which provides effective FPGA resource utilization. The derived buffer solution is described by the VHDL language and can be used effectively in any hardware project.

Methods for the buffer design

The methods of the data buffer design evolved for decades. The memory bandwidth increase is the usual goal of the buffer design. The easiest way to increase the memory bandwidth is to have multiple memory blocks in parallel. Similarly, it is possible to implement a memory with an extra-large data word length that stores several adjacent data. But in these cases, when the memory is out of FPGA, in addition to several memory chips, it is necessary to have many separate outputs from FPGA for addresses and data, which is often unacceptable. The impact of this problem is somewhat reduced by organizing several blocks of cache memory in FPGA. By dividing the address space into multiple banks, using one memory bank for odd addresses and another one for even addresses, the adjacent addresses can be accessed simultaneously. For example, four banks can be used to access four pixels in a 2×2 block. In addition, for efficient access to the pixels in the aperture, the address can be coded as proposed in [3].

In the pipelined random access to RAM, one process can write results to one memory bank, and another can read data from the second bank. When the processing of the next data array is completed, the banks switch their roles. At the same time, a third memory bank is used for better synchronization

[4]. But such switching of banks adds a long period of time to the latency of the algorithm and has the consequence of increasing the hardware costs of the system, and the use of more FPGA pads.

A more practical approach is to run the memory at a higher clock frequency than the rest of the system. Double data rate (DDR) memory is one example of memory that allows data to be transferred twice per clock. As a rule, modern high-capacity FPGAs have dedicated outputs and a built-in access controller for external dynamic DDR memory of recent generations [5]. At the same time, the project simulates multiport memory due to access time slots. In addition, blocks of the buffer memory are required for writing and reading, since dynamic memory has high throughput only when transferring rows of data from neighboring cells. Unfortunately, in many projects, DDR memory is also required to support the operating system of the processor embedded in the FPGA, and therefore the bandwidth of this memory drops when processing large data arrays.

Pipeline and FIFO first-in-first-out (FIFO) type buffers are two popular types of memory organization methods utilized in FPGA. They are distinguished in the following. The data stored in and retrieved out of a pipeline is also in the first-in-first-out category. But the steps of storage and retrieval are constant as in the serial-in-serial-out shift register. A FIFO buffer is a storage where the data can be pushed into and popped out with the same data order, but these operations can be uncorrelated. However, the implementation of both long pipeline buffer and FIFO is based on RAM which is operating as the circular buffer. The method of such buffers design is explained in [6]. But the designer must organize the proper order of data pushing and popping separately.

If the data are executed sequentially, then it is worth using the buffers of the FIFO type, which cell groups store blocks of data, and the output data are selected by the local addresses [7].

When the algorithm can be represented by some Petri net, then the stream processing computational model can be used. In this model, the computational node or processing kernel consists of the stencil buffer and computing module connected to the buffer inputs. During the computational process, the input data are loaded into the buffer asynchronously and just when they form the proper stencil the computations start [8]. So, the buffer is really the register pipeline with a large set of outputs, which is often the inefficient solution.

When the usual serial program is mapped in the hardware, then the data buffer with the last-in-first-out discipline is needed. The method of such stack buffer design as well as the respective finite state machine development named Hierarchical Finite State Machines (HFSM) method is described in [9].

The von Neumann architecture paradigm is widely used in which each datum has its own robust address in the common address space. The data buffers are implemented as the cache memory blocks in this paradigm. Note, that in particular, when the data lose their addresses in the moment before their execution, then this cache memory can be represented as a usual FIFO buffer. Therefore, the usual data buffer is often called the cache [10]. The method of the cache buffer design for FPGA based on the optimized data throughput is described in [11]. When the FPGA application deals with dynamic memory allocation, then the cache buffers can be designed using the method of algorithm analysis which selects the independent and shared memory fields [12].

The dataflow processing is the kind of algorithm that is usually implemented in FPGA because the FPGA architecture provides the effective implementation of such algorithms. The most common model for the dataflow algorithm representation is the Kahn processing network (KPN). The nodes of this network represent the operations or actors, and the edges represent the dataflows. The edges contain the FIFO buffers of the proper length. Usually, KPN is mapped into FPGA by one-to-one mapping. So, the FIFO buffers serve as the proper data buffers [13]. Note, that this model considers that the data are retrieved from FIFO in arbitrary order, i. e., the buffer can contain several outputs from its head registers.

The unified modeling language or UML provides an effective KPN representation. Many tools like IBM Rational Rhapsody provide translation of the UML description into hardware [14]. The Matlab Real-Time Workshop (RTW) tool offers code generation capabilities directly from Simulink graphical system descriptions which is a kind of KPN [15]. These tools implement the FIFO buffers as they are foreseen in the given KPN. But these buffers must obey the rules of the asynchronous reading and writing data in them in the respective order.

The synchronous dataflow (SDF) graph is the abridged KPN model, in which all dataflows are synchronous. Note, that two dataflows are synchronous if the data in one flow are correlated with the data in the other one, for example, both data samples have the same index sets. The FIFO buffers in the SDF model are always synchronous ones, and this model is usually free of deadlocks. This model gives simple mapping to hardware, providing effective methods of structure optimization like pipelining, retiming, folding, and resource sharing [16]. This idea is expanded and fulfilled in the SDF modeling framework Ptolemy [17]. By this method, the optimized data buffers are synthesized as well. But the synthesis results can be far from excellent because the optimization is performed by hand or automatically. Through this process, the effective schedule is searched which disagrees with the hardware minimization.

When the algorithm given by SDF has no loops and feedback then it is usually represented by the dataflow graph (DFG). Then, the data buffers with the minimum volume can be synthesized using the method proposed in [18]. This method combines the register allocation by the left-edge scheduling and the SDF folding.

When the 2D signals or images are processed, then the problem of the buffer design becomes more complex. In this situation, the multidimensional SDF can be used, in which the data have the vectors of indexes which can be considered as the pixel coordinates in the image frame [19]. But the buffer design remains a complex task.

Many algorithms including ones of image processing are represented by the loop nest. The index vectors of the loop nest iterations and the data themselves form the multidimensional grid, and the algorithm does the respective lattice-like DFG. The method of the systolic processor design is widely used for mapping these algorithms both into the processor structure and into the timetable of the operator execution [16]. The pipelined data buffers are the obligatory result of such mapping. Therefore, this method is widely used now to design data buffers in many synthesis methods and automatic design frameworks.

Placing the operators in the iteration space and mapping them in the structure and timetable is used in [20] as well. To optimize the data buffers, the system of linear inequalities which takes into account the operator data dependencies, data moving delays, and time limitations. This system is solved using the usual integer linear problem solver. As a result, the throughput is optimized and the pipelined data buffers are synthesized. But the synthesis process becomes very complex when the problem dimension increase.

This method is expanded using the polyhedral model of the parallel algorithm DFG representation and its mapping [21]. Due to this method, the executed iterations of the algorithm and their data form the polyhedron in the multidimensional iteration space which limits the volume of the lattice-like DFG. Each iteration in it occupies a particular integer vector in the space. This polyhedron is mapped into the systolic structure of the computer and the timetable using the optimized affine transformations of this space. When the loop nest describes the data array behavior then the result of the mapping is a set of pipelined data buffers. A similar method is proposed in [22]. The method named lattice-based partitioning is based on the same principle and performs the selection of a set of distributed buffers [23].

FPGA hardware is utilized very well providing high throughput when the data are reused frequently. The method of the buffer design described in [24] provides the data reusing when the algorithm performs the sequential array processing using the modulo addressing. A more sophisticated method utilizing data reuse is proposed in [25]. The approach of the systolic processor design is implemented in it and the data which are fetched from the one- or two-dimensional array are reused in the algorithm.

The buffers of different lengths should be designed for different data array sizes. It is proposed to use a universal buffer, which is adjusted to the array size and the computed frame in it with the possibility of dynamic reconfiguration [26]. A similar method for image processing is described in [27], which is capable of transposing the position of pixels in the frame, as well as performing image correction at the frame edges.

The works [28, 29] present general methods of designing a pipelined structure for image processing with a sliding aperture selected for processing. At the same time, the functions that are sequentially performed in the algorithm are mapped in the corresponding processing blocks, which are separated from each other by buffer blocks that store several adjacent lines. The interconnections between processing blocks and buffer blocks are buses that correspond to the edges of DFG.

The smart buffer is a compiler-generated data buffer that provides re-using the fetched data in the sliding aperture. The structure of the buffer is determined by the window size, array size, and the stride of the reuse in each dimension [30]. This method is effectively utilized in the Riverside optimizing compiler for configurable computing (ROCCC) approach and compiler [31].

Goals of the investigation

The analysis of different methods of the data buffer design makes it possible to conclude the following.

KPN mapping gives a set of pipelined data buffers in a natural manner. However, the resulting buffers have several output ports in many cases and the deadlock problem is solved hard.

SDF is the abridged model of KPN, but it is a rather impressive one and it is free of deadlocks. Many dataflow algorithms like digital signal processing are represented as SDF and are effectively mapped into hardware structures including pipelines and FIFOs.

The most sophisticated and formalized methods are ones that are based on the representation of the algorithm as DFG in the multidimensional grid and mapping it into the systolic-like processor structures. Many of them are implemented in high-level synthesis frameworks. But these methods are limited by the algorithms which are represented by the loop nests and do not take into account the features of the hardware technology.

The goal of the investigation is to develop a new method of data buffer design that is more sophisticated and is able to take into account the features of the FPGA architecture. The method is intended for the pipeline buffer design however it is fitted for the buffers based on RAM. These buffers are designed in general for the streaming algorithms like DSP, image processing, or others that can be represented by SDF.

The derived buffers must be optimized both in the clock frequency and in hardware. Therefore, first of all, the FPGA features are considered. Then, the method of the pipelined datapath design is selected which involves the better features of the methods considered above. And next this method is adapted to the data buffer design.

FPGA resources for the buffer design

The FPGA chip usually contains sufficient volume of different memory resources. Usually, the basic building block is the Look-Up Table (LUT) in Xilinx FPGAs or Adaptive Logic Module (ALM) in Intel FPGAs. Each of them is accompanied by one or two 1-bit registers. These registers usually form the storage elements of the pipeline stages including the pipeline buffers. LUT by itself is configured as the buffer RAM with a volume of up to 64 bits, and with several possible reading ports. Moreover, it can be configured as the pipeline buffer of the variable length. Fig. 1 illustrates the structure of such an SRL16 primitive which contains the 16-bit shift register, and each of its taps is selected statically or dynamically by the output multiplexor.

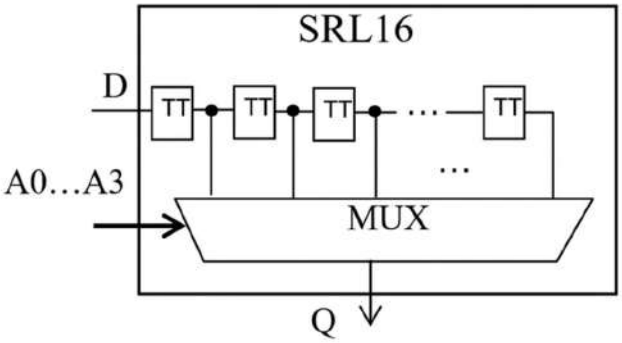


Fig. 1. Pipeline buffer SRL16 structure

FPGA contains from tenths to thousands of two-port blocked RAMs (BRAMs). Each of them contains kilobytes of memory of programable bit width. The ratio of BRAM number to LUT number in FPGA is equal from 60 to 200. Usually, they can be configured as FIFO buffers [32, 33].

The Intel Hyperflex FPGA architecture provides the pipeline buffers of the arbitrary length in the routing segments in the inter ALM communications. These buffers enable the highest clock frequencies in Intel Stratix® 10 and Intel Agilex™ devices [34].

Usually, the most effective structure solutions are derived from the register transfer level (RTL) design. But in such a design, the buffer selection, and its dynamic control, which depends on the modules attached to it, is a hard design task. Therefore, the usual solution is selection the FIFO buffer based on BRAM, which takes increased hardware volume. The SRL16 buffers are utilized rarely in some specific finite state machines (FSMs), filters, or encryptors [35]. The Hyperflex register utilization in the projects takes specific knowledge about the SDF optimization and is not fulfilled in most cases when SDF contains the loops [34].

Spatial SDF method

A method of designing the pipelined datapaths by mapping SDF is proposed in [36, 37]. The feature of the method is that SDF is represented in the resource-time space in the form of an algorithm configuration (AC). The method makes it possible to search for a schedule, minimize the number of processor units (PUs), and search for effective interprocessor connections simultaneously. Here, PU means an elementary computing element with or without result registers, for example, an adder, a multiplier with a register, a pipeline buffer, etc. Therefore, it makes sense to create a method for the data buffers development based on this method. It is described below in short.

At the first stage of the synthesis, according to the specified method, operators-nodes of a homogeneous SDF together with the data dependency edges are located in three-dimensional space $\mathbb{Z}^3$ as sets of vectors K_i and D_j, respectively, taking into account the conditions, given in [36]. The coordinates of the vector $K_\mathrm{i} = (s,q,t)^\mathrm{T}$ mean the number s of the PU, where the operator is executed, the type q of this PU, and the time component t, which is equal to the clock number during the execution of the algorithm. Vectors K_i with equal time components form one row and are executed simultaneously. The time component $R(D_\mathrm{j})$ of the vector $D_j = K_i - K_l$ is equal to the delay between the executions of operators whose nodes K_i, K_l are adjacent. The number of PUs is minimized by fulfilling the requirements $|K_{s,q}|\rightarrow L$, i.e. the number of nodes mapped in the s-th PU approaches to L, where L is the algorithm execution period in clock cycles. In addition, when forming the effective algorithm configuration, it is desirable to build a perfect spanning tree of SDF, as suggested in [38].

In the second step, AC is balanced, which consists in adding delay nodes to the edges of SDF until the time components of all vectors D_j are equal to 0 or 1. After that, AC is optimized by permuting the node vectors from the same column in order to minimize the number of registers and the number of multiplexer inputs in the resulting structure and/or using other strategies, for example, retiming. Also, the number of registers is minimized by gluing delay nodes from the same column that store the same operand.

In the third step, the obtained optimized AC is mapped in the graph of the computer structure in the subspace $\mathbb{Z}^2$ named as the structure configuration. This is done by gluing the node vectors with the same coordinates s, and q. AC is transformed into the schedule of operator execution, using the property that the time component of the vector K_i is equal to the moment of execution of the operator, regardless of the number of the execution period. At the same time, the resulting structure is not built and the schedule is not formed because the resulting structure is described in VHDL on the base of information in AC.

Method for the buffer design

Consider AC $C'_{\mathrm{A}v}$ which performs the iterative algorithm with the period of $L = 4$ clock cycles, and which consists only of input and output nodes. This AC is mapped into the data buffer. When placing the nodes of CA in the space $\mathbb{Z}^3$, one should use some strategies to minimize the number of connections between PUs. The location of the nodes of the delay operators according to the strategy of placing the edges $D_{i,j}$ in parallel to the axis Ot in the second step of the synthesis is shown in Fig. 2,

a. And the configuration C'_{Av} according to the strategy of placing the edges $D_{i,j}$ at an angle to the axis *Ot* is shown Fig. 2, b. The structure configurations corresponding to these CAs are shown in Fig. 2, c, and 2, d, respectively. Here, the bold points mean the nodes of input-output or some operator nodes, and circles mean the delay nodes mapped into the registers. The bars mean the multiplexers attached to the PU inputs, which perform the selection of the operand when it is read from the respective register.

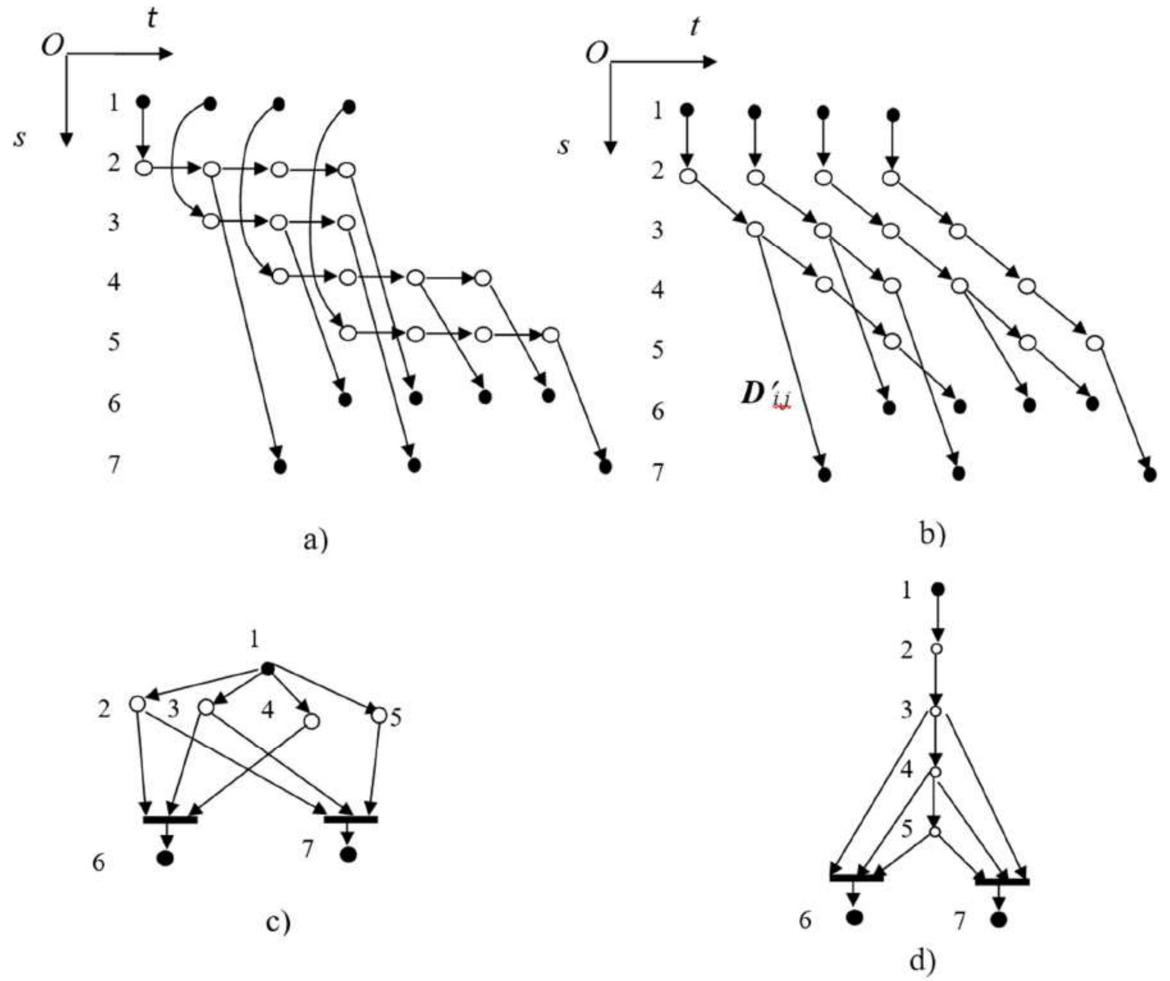


Fig. 2. AC which edges placed according to the strategy of RAM (a) or pipeline buffer (b) synthesis, and respective RAM (c) and FIFO (d) configurations

Analysis of these structure configurations shows that they correspond to two-port RAM (one port to read-write, second one only to read) and pipelined data buffer, respectively. Applying one or another strategy of connection number minimizing, the designer can orient the process of synthesis of the data buffer to implementation in the form of RAM or a register pipeline. The strategy should be chosen taking into account the following features.

When synthesizing the buffer based on RAM, the variable x_i is allocated in the respective register, i.e. the chain of delay nodes is located on a straight, which is parallel to the axis *Ot*. Also, one register is assigned to several variables whose periods of existence do not overlap, i.e. several chains of delay nodes are located on a straight, which is parallel to the axis *Ot*, and these chains do not overlap. At this process, the edges $\boldsymbol{D}_{i,j}$, which are adjacent to the outputs of the edges $\boldsymbol{K}_{i,j}$ of the AC before balancing the relation

$$\max_{i,j}(t_{Di,j}) \le L \qquad (1)$$

is satisfied, where $t_{Di,j}$ is the time component of the vector-edge $\boldsymbol{D}_{i,j}$. If it is not observed, it is necessary to cut the balanced AC C'_{Av} into several subconfigurations, each of which will corresponds to its own

RAM or ensure overwriting of the variable x_i for which it is not observed the inequality (1), in the second register of the RAM after L clock cycles. It is obvious that the volume of the resulting RAM for AC C'_{Av} with λ input nodes (bold points in Fig. 2) is equal to

$$N_P = \lambda. \tag{2}$$

When the pipeline buffer is designed, then the variable x_i is sent to the adjacent pipeline register in each clock cycle and, passing through a chain of $t_{Di,j}$ registers is outputted from it to the input of PU which receives this variable. This is equivalent to the fact that the chains of adjacent nodes $\boldsymbol{K}_{i,j}$ of the delay operators at uniformly increasing coordinates $s_{i,j}$, and $t_{i,j}$ are placed along parallel lines, located at an angle to the axis Ot (Fig. 2, b). Therefore, the value of $t_{Di,j}$ in (1) can be any, however, to minimize the number of the register pipeline stages, the number of different values of the vectors $\boldsymbol{D}'_{i,j}$ must be minimal. The number of registers in the pipeline is equal to

$$N_P = \max_{i,j}(t_{Di,j}). \tag{3}$$

Thus, AC which performs the data transfer between input and output ports after its balancing and optimization according to one of two strategies gives a minimized amount of memory in the resulting data buffer. We get a buffer structure with memory organized in the form of RAM or a register pipeline. At the same time, the number of registers in RAM is smaller than in the pipeline of registers, if the number of input nodes that are mapped to one port node (the number of different variables entering one PU) in AC is less than the maximum delay of the variable that is calculated in this PU, i.e. at

$$\lambda = \max_{i,j}(t_{Di,j}). \tag{4}$$

When the resulting pipelined buffer is performed in the SRL16 primitive, then the method must take into account the fact that it has a single output (see Fig. 1). This adds the additional limitation to AC placement in the space that only a single edge must connect any delay node with the node which is mapped into the output port PU. AC in Fig. 2, b does not satisfy this condition. Therefore, it is split into two subconfigurations in Fig 3, a, which satisfies it and is mapped into the structure with two units implemented in SRL16 primitives (Fig 3, b).

The SRL16 primitive has an additional clock enable input, the control of which makes it possible to slow down the data moving through the pipeline registers. When using this input, the number of registers can be minimized if the value of $R(\boldsymbol{D}_j)$ is greater than the number of available registers in the pipeline. Fig. 4 shows an example of the transformation of AC, shown in Fig. 3, a, for the purpose of additional delay of the operands. Such delays correspond to the vectors $\boldsymbol{D}_j$, which are placed parallel to the axis Ot. Note, that the number of nodes that have the same coordinate s must not be higher than the computation period L.

The Fig. 4 analysis shows that the technique of the clock enable control allows us to minimize both the pipeline register number and output multiplexers substantially. This is important when the pipeline registers are performed on the base of usual registers because it saves hardware and minimizes the clock period.

If the nodes-sources of considered AC have different spatial coordinates s (in the examples above s = 1), then an input multiplexor is obtained at the input of the SRL16 primitive. To minimize such multiplexers, the method can be used which is described in [39].

Thus, the method of designing the pipelined datapaths with buffers based on SRL16 primitives looks like the following. The initial data are AC, algorithm execution period L, and other optimization parameters. The method is performed in the same way as described in [37, 38], with the exceptions described below.

In the first stage of synthesis, the AC subgraphs corresponding to the transfer of operands between computer resources with time delays and/or shuffling of operands, which are expected to be mapped into separate data buffers, should be selected.

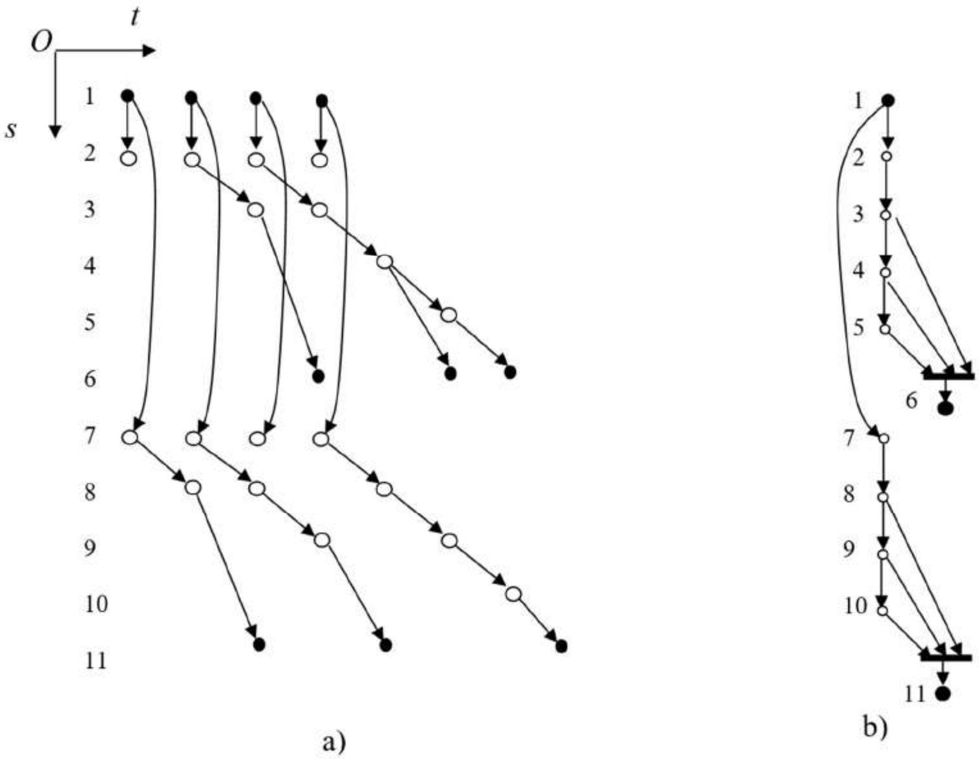


Fig. 3. AC which is split to AC in Fig. 2, b (a) and its mapping into SRL16 structures (b)

In the first stage of synthesis, the AC subgraphs corresponding to the transfer of operands between computer resources with time delays and/or shuffling of operands, which are expected to be mapped into separate data buffers, should be selected.

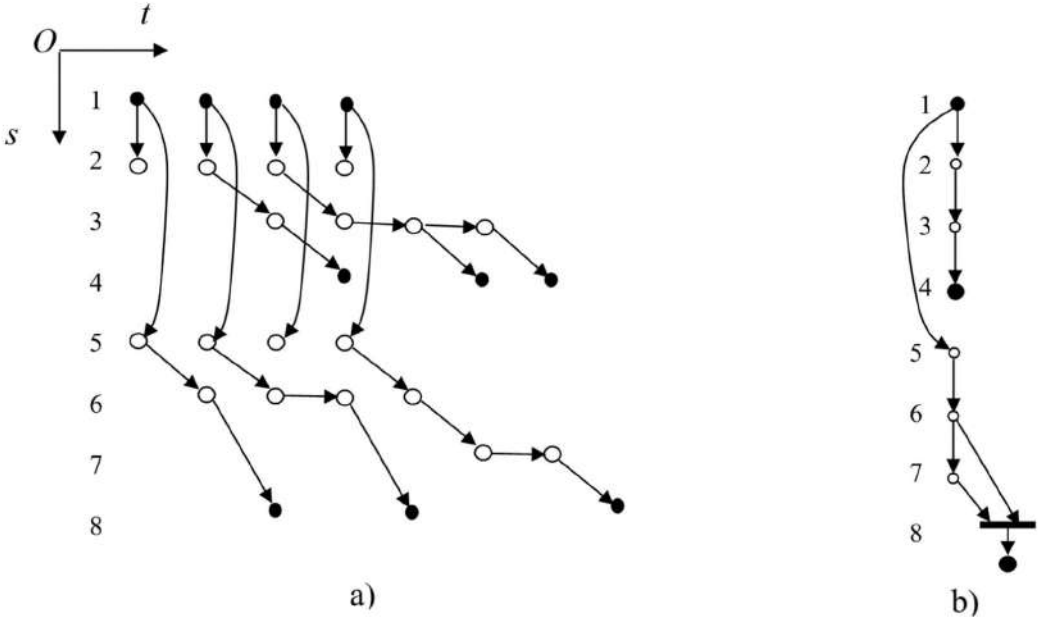


Fig. 4. Modified AC Fig. 3, a (a), and its mapping into SRL16 structures (b)

In the second stage, it is necessary to balance the dependence edges using the intermediate delay nodes. The number of intermediate delay nodes is minimized, if possible. The delay nodes are placed on parallel lines that are at an angle to the time axis or parallel to this axis in such a way that adjacent delay nodes differ in time coordinates by one beat. The requirements for the correct placement of nodes are fulfilled, including the requirement to implement a buffer with one input and one output. If it is impossible to get a single input in the buffer, the heuristic of minimizing the number of inputs of the additional multiplexer at the buffer input is used according to [39], and if it is impossible to receive a buffer with one output, the chain of delay nodes is split so that they are mapped in additional buffers (see Fig. 4).

The dependency edges together with the corresponding delay nodes which are incident to the nodes consuming the buffered data should be mapped in the data buffer. When a control algorithm is designed, if only edges are displayed in the buffer that is at an angle to the time axis, then operands are written to the buffer in each clock cycle. If there are edges that are parallel to this axis, then writing to the buffer is prohibited in the corresponding clock cycles (see Fig. 4).

At the third stage, the pipelined datapath is described in VHDL according to the method presented in [38] and is compiled into an FPGA configuration that contains the buffers based on SRL16 primitives, which correspond to the selected AC subgraphs.

Experimental results

Consider the design of the input buffer for the pipelined datapath performing the 8-point discrete cosine transform (DCT). The DFG of this algorithm is often based on the Chen algorithm [41]. This algorithm is distinguished in that its period of the pipelined computations is equal to L = 8 clock cycles, eight input data of a single DCT transform need to be delayed and permutated in the input buffer before their calculations. DFG of the first stage of this algorithm which needs the data buffer is shown in Fig. 5.

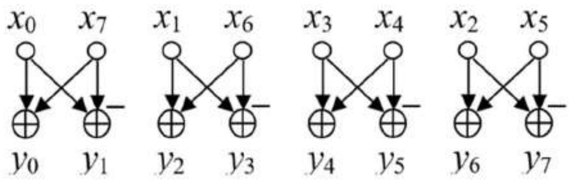
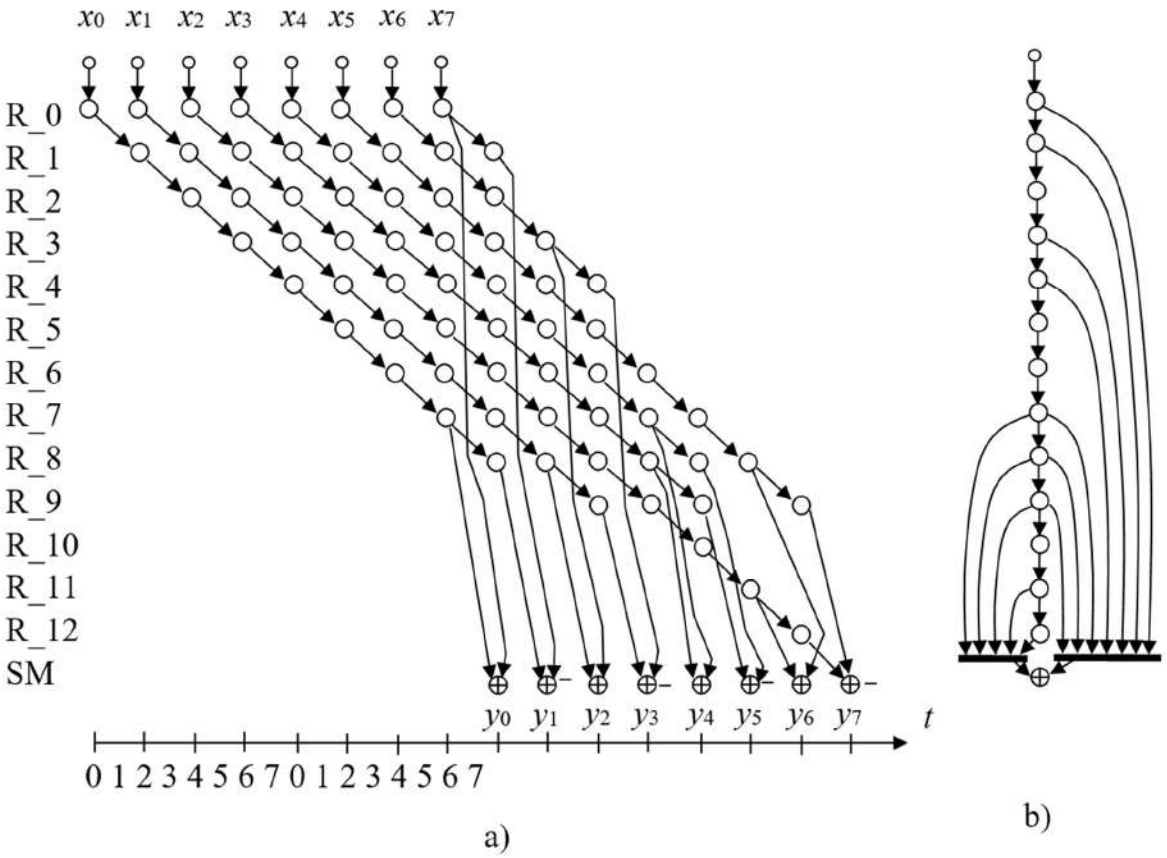


Fig. 5. DFG of the first stage of the DCT algorithm

Optimized AC which is mapped into pipelined buffer and adder, and respective structure configuration are illustrated in Fig. 6. Here, the resource names are placed in the *Os* axis and the clock cycle number modulo L = 8 is mapped in the axis *Ot*. The addition-subtraction operator node has the plus sign. This AC is described in VHDL as follows.

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.Numeric_STD.all;
entity DCT_BUF is
       port(
             CLK : in STD_LOGIC;
             RST : in STD_LOGIC;
             START : in STD_LOGIC;
             X : in SIGNED(8 downto 0);
             Y : out SIGNED(8 downto 0)
       );
end DCT_BUF;
architecture synt of DCT_BUF is
       type TARRAY16 is array (0 to 15) of SIGNED(8 downto 0);
       type TN is array(0 to 7) of natural range 0 to 15;
       constant al: TN:=(7,8,8,9,8,9,11,12);
       constant ar: TN:=(0,1,3,4,7,8,8,9);
       signal r1,r2:TARRAY16; -- register array of SRL16
       signal cycle:natural range 0 to 7;
       signal sm,l,r: SIGNED(8 downto 0);
begin
       CT8:process(CLK) begin -- period counter
             if CLK'event and CLK='1' then
                   if START='1' then
                         cycle<=0;
                   else
                         cycle<= (cycle+1) mod 8;
                   end if;
             end if;
       end process;

 l<= r1(al(cycle));
       r<= r2(ar(cycle));
       SRL16_BUF:process(CLK) begin -- SRL16 description
             if CLK'event and CLK='1' then
                   r1<=X & r1(0 to 14); -- FIFO shift
                   r2<=X & r2(0 to 14); -- FIFO shift
                   case(cycle) is
                         when 0|2|4|6 => sm<= l + r; -- adder
                         when others => sm<= l - r; -- subtractor
                   end case;
             end if;
       end process;
       Y<=sm;
end synt;
```



x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7
R_0
R_1
R_2
R_3
R_4
R_5
R_6
R_7
R_8
R_9
R_10
R_11
R_12
SM
-
y_0 y_1 y_2 y_3 y_4 y_5 y_6 y_7
t
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
a)
b)

Fig. 6. Balanced spatial SDF for DFG in Fig. 6 (a), and respective structure configuration (b)

Here, signals *r*1, *r*2 represent two pipeline register chains, which load the input data *X* in each clock cycle. They are synthesized after splitting AC in Fig. 6, a in two subconfigurations like it is done in Fig. 4. The signals from them *l*, *r* are read at addresses which are sampled from ROMs *al*, *ar*. These signals are directed to the left and right inputs of the adder-subtractor with the register *sm* deriving the result *Y*. The calculating period counter cycle counts modulo *L* = 8 and controls both the sign of the adder *sm* and the pipeline register chains through the ROMs *al, ar*.

This project is compiled by the Xilinx ISE and Vivado CAD packages into FPGAs of different series. The results of compilations are shown in Table 1.

This table analysis shows that the ISE synthesizer recognizes the template of the SRL16 primitive and the synthesis results are the data buffers with the minimum hardware volume and good performance. The Vivado synthesizer first tries to compound both pipeline buffer branches into one and then minimizes the trigger number by substituting the chains of registers with the SRL16 primitives. The inferred structure is illustrated in Fig. 7. One can see, that additionally, the synthesizer doesn`t perform the resource sharing of the adder-subtractor. As a result, the hardware volume in the register number is much higher.

Table1.

Results of a configuration of the buffer project in FPGAs

FPGA series	Compiler	Slice Flip Flops	LUTs	LUTs used as logic	LUTs used as SRL16	Minimum clock period, ns
Virtex-4	ISE 14.7	12	37	19	18	3.14
Spartan-3A	ISE 14.7	12	37	19	18	5.47
Spartan-6	ISE 14.7	12	39	29	10	4.72
Artix-7	Vivado2016	111	44	39	5	4.06

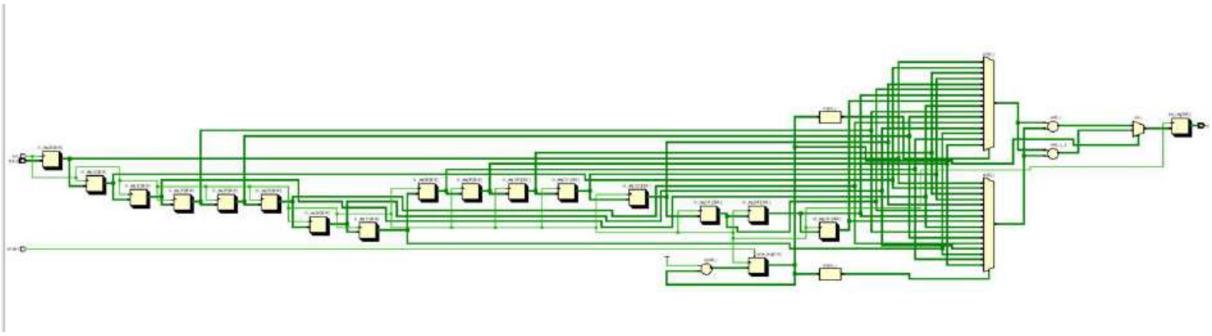


Fig.7. Data buffer structure derived by the Vivado design tool

Synthesis framework

As one can see from the method description and the design example, the considered algorithm is given in the graphical form effectively. For the design method investigations, the synthesis framework is developed named SDFCAD [42]. The framework is able to perform the graphical input of SDF of the DSP algorithms with the given period *L* and data bit width. SDF can be optimized either manually or automatically using one of the genetic programming algorithms [40]. One of two strategies of the buffer design are used by the optimization as well. In particular, the pipelined buffers for the DCT processor are synthesized automatically very well [42].

Conclusions

A new method of the data buffer design is proposed, which is intended for the complex pipelined datapaths development and configuring in FPGA. The method is based on the SDF representation in the three-dimensional space, optimization them and describing in VHDL. Depending on the optimization method the derived buffer is based either on RAM or on the register pipeline. The feature of the method consists in that the pipeline buffer is inferred into the SRL16 primitives of the AMD-Xilinx FPGA series which substantially saves the hardware. The method is built in the experimental SDFCAD framework intended for the pipelined datapath synthesis.

References

[1] D. Koch, F. Hannig, and D. Ziener, *FPGAs for Software Programmers*. Springer, 2016, p. 327.

[2] Uwe Meyer-Baese, *Digital Signal Processing with Field Programmable Gate Arrays*. Heidelberg: Springer Berlin, 2014, p. 930.

[3] K. Kim and P. Kumar, "Parallel memory systems for image processing," in *Proceedings CVPR '89: IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 654–659. doi: https://doi.org/10.1109/CVPR.1989.37915.

[4] S. Khan, D. Bailey, and G. S. Gupta, "Simulation of Triple Buffer Scheme (Comparison with Double Buffering Scheme)," in *2009 Second International Conference on Computer and Electrical Engineering*, pp. 403–407. doi: https://doi.org/10.1109/ICCEE.2009.226.

[5] S. Churiwala, *Designing with Xilinx® FPGAs : Using Vivado*. Switzerland: Springer, 2017.

[6] Hartmut F.-W. Sadrozinski and J. Wu, *Applications of Field-Programmable Gate Arrays in Scientific Research*. CRC Press and Taylor & Francis, 2011, p. 144.

[7] P. Sedcole, K. Cheung, G. A. Constantinides, and W. Luk, “RunTime Integration of Reconfigurable Video Processing Systems,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 9, pp. 1003–1016, doi: https://doi.org/10.1109/TVLSI.2007.902203.

[8] K. Sano and H. Nakahara, “Hardware Algorithms,” in *Principles and Structures of FPGAs*, H. Amano, Ed., Springer Singapore, 2018, pp. 137–177. doi: https://doi.org/10.1007/9789811308246_6.

[9] V. Sklyarov, I. Skliarova, A. Barkalov, and L. Titarenko, *Synthesis and Optimization of FPGA-Based Systems*. Cham Springer International Publishing, 2014, p. 432.

[10] D. G. Bailey, *Design for Embedded Image Processing on FPGAs*. John Wiley & Sons, 2011, p. 482.

[11] R. Sass and A. G. Schmidt, *Embedded Systems Design with Platform FPGAs*. Morgan Kaufmann, 2010, p. 389.

[12] F. Winterstein, K. Fleming, H.-J. Yang, S. Bayliss, and G. Constantinides, “MATCHUP: Memory abstractions for heap manipulating programs,” presented at the FPGA ’15: Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, Monterey, California, USA: Association for Computing Machinery, 2015, pp. 136–145. doi: https://doi.org/10.1145/2684746.2689073.

[13] R. Woods, J. McAllister, G. Lightbody, and Y. Yi, *FPGA-based implementation of signal processing systems*, 2nd Ed. Hoboken, Nj : Wiley, 2017, p. 448.

[14] L. Granado and O. Berreteaga, “Creating Rich Human-machine Interfaces with Rational Rhapsody and Qt for Industrial Multi-core Real-time Applications,” *Procedia Manufacturing*, vol. 3, pp. 1903–1909, 2015, doi: https://doi.org/10.1016/j.promfg.2015.07.233.

[15] J. Hwang, B. Milne, N. Shirazi, and J. D. Stroomer, “System Level Tools for DSP in FPGAs,” in *FieldProgrammable Logic and Applications*, G. Brebner and R. Woods, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 534–543.

[16] K. K. Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation*. Wiley, 1999, p. 784.

[17] E. A. Lee, S. Neuendorffer, and M. Wirthlin, “Actor-Oriented Design of Embedded Hardware and Software Systems,” *Journal of Circuits System and Computers*, vol. 12, no. 03, pp. 231–260, Jun. 2003, doi: https://doi.org/10.1142/s0218126603000751.

[18] M. Ruvald Pedersen and J. Madsen, “Optimal register allocation by augmented leftedge algorithm on arbitrary controlflow structures,” in *NORCHIP 2012*, pp. 1–6. doi: https://doi.org/10.1109/NORCHP.2012.6403107.

[19] P. K. Murthy and E. A. Lee, “Multidimensional synchronous dataflow,” *IEEE Transactions on Signal Processing*, vol. 50, no. 8, pp. 2064–2079, doi: https://doi.org/10.1109/TSP.2002.800830.

[20] J. Cong, W. Jiang, B. Liu, and Y. Zou, “Automatic memory partitioning and scheduling for throughput and power optimization,” presented at the IEEE/ACM International Conference on Computer-Aided Design, San Jose, California: Association for Computing Machinery, 2009, pp. 697–704. doi: https://doi.org/10.1145/1687399.1687528.

[21] Y. Wang, P. Li, and J. Cong, “Theory and algorithm for generalized memory partitioning in high-level synthesis,” presented at the FPGA ’14: Proceedings of the 2014 ACM/SIGDA International Symposium on Field-programmable Gate arrays., Monterey, California, USA: Association for Computing Machinery, 2014, pp. 199–208. doi: https://doi.org/10.1145/2554688.2554780.

[22] J. Cong and J. Wang, “PolySA: PolyhedralBased Systolic Array AutoCompilation,” in *2018 IEEE/ACM International Conference on ComputerAided Design (ICCAD)*, pp. 1–8. doi: https://doi.org/10.1145/3240765.3240838.

[23] L. Gallo, A. Cilardo, D. Thomas, S. Bayliss, and G. A. Constantinides, “Area implications of memory partitioning for highlevel synthesis on FPGAs,” in *2014 24th International Conference on Field Programmable Logic and Applications (FPL)*, pp. 1–4. doi: https://doi.org/10.1109/FPL.2014.6927417.

[24] Y. Wang, P. Zhang, X. Cheng, and J. Cong, “An integrated and automated memory optimization flow for FPGA behavioral synthesis,” in *17th Asia and South Pacific Design Automation Conference*, pp. 257–262. doi: https://doi.org/10.1109/ASPDAC.2012.6164955.

[25] Z. Guo, W. Najjar, and B. Buyukkurt, “Efficient hardware code generation for FPGAs,” *ACM Trans. Archit. Code Optim.*, vol. 5, Art. no. 1, 2008, doi: https://doi.org/10.1145/1369396.1369402.

[26] R. Shi, J. S. J. Wong, and H. K. H. So, “HighThroughput Line Buffer Microarchitecture for Arbitrary Sized Streaming Image Processing,” *Journal of Imaging*, vol. 5, no. 3, 2019, doi: https://doi.org/10.3390/jimaging5030034.

[27] D. G. Bailey and Ambikumar, Anoop S, “Border Handling for 2D Transpose Filter Structures on an FPGA,” *Journal of Imaging*, vol. 4, no. 12, 2018, doi: https://doi.org/10.3390/jimaging4120138.

[28] Y. Ikarashi, J. RaganKelley, T. Fukusato, J. Kato, and T. Igarashi, “Guided Optimization for Image Processing Pipelines,” in *2021 IEEE Symposium on Visual Languages and HumanCentric Computing (VL/HCC)*, pp. 1–5. doi: https://doi.org/10.1109/VL/HCC51201.2021.9576341.

[29] M. A. Özkan, O. Reiche, F. Hannig, and J. Teich, “FPGAbased accelerator design from a domainspecific language,” in *2016 26th International Conference on Field Programmable Logic and Applications (FPL)*, pp. 1–9. doi: https://doi.org/10.1109/FPL.2016.7577357.

[30] Z. Guo, W. Najjar, and B. Buyukkurt, “Efficient hardware code generation for FPGAs,” *ACM Trans. Archit. Code Optim.*, vol. 5, Art. no. 1, 2008, doi: https://doi.org/10.1145/1369396.1369402.

[31] W. A. Najjar, J. Villarreal, and R. J. Halstead, “ROCCC 2.0,” in *FPGAs for Software Programmers*, D. Koch, F. Hannig, and D. Ziener, Eds., Cham: Springer International Publishing, 2016, pp. 191–204. doi: https://doi.org/10.1007/9783319264080_11.

[32] “UltraFast Design Methodology Guide for the Vivado Design Suite (v2013.3).” Accessed: Oct. 23, 2013. [Online]. Available: www.xilinx.com

[33] “7 Series FPGAs Memory Resources User Guide. UG473 (v1.14).” Accessed: Jul. 03, 2019. [Online]. Available: www.xilinx.com

[34] “Intel® Hyperflex^{TM} Architecture High Performance Design Handbook,” Oct. 2021. Available: https://www.intel.com/programmable/technical-pdfs/683353.pdf

[35] J. Chu and M. Benaissa, “Low area memoryfree FPGA implementation of the AES algorithm,” in *22nd International Conference on Field Programmable Logic and Applications (FPL)*, pp. 623–626. doi: https://doi.org/10.1109/FPL.2012.6339250.

[36] A. Sergiyenko, O. Maslennikow, and Y. Vinogradow, “Tensor approach to the application specific processor design,” in *2009 10th International ConferenceThe Experience of Designing and Application of CAD Systems in Microelectronics*, pp. 146–149.

[37] A. Sergiyenko, A. Serhienko, and A. Simonenko, “A method for synchronous dataflow retiming,” in *2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, pp. 1015–1018. doi: https://doi.org/10.1109/UKRCON.2017.8100404.

[38] A. M. Sergiyenko and V. P. Simonenko, “Method of synchronous dataflow scheduling,” *System research and information technologies*, no. 1, pp. 51–62, Mar. 2016, doi: https://doi.org/10.20535/srit.2308-8893.2016.1.06.

[39] A. M. Sergiyenko and V. P. Simonenko, “Otobrazenie perioditsheskich algorithmov w programmiruemye logitsheskie integralnye schemy,” *Electronic Modeling*, vol. 29, no. 2, pp. 49–61, 2007.

[40] A. Sergiyenko, A. Serhienko, and V. Romankevich, “Genetic Programming of Pipelined Datapaths for FPGA,” in *2020 IEEE 40th International Conference on Electronics and Nanotechnology (ELNANO)*, pp. 802–806. doi: https://doi.org/10.1109/ELNANO50318.2020.9088773.

[41] A. Sergiyenko, A. Serhienko, and V. Romankevich, “Genetic Programming of Discrete Cosine Transform Processors,” presented at the 6-th International Conference on High-Performance Computing (HPC-UA 2020), 2020, pp. 1–6.

UDC 004.056.5 *DOI: https://doi.org/10.20535/2708-4930.3.2022.265480*

ORGANIZATION OF FAST EXPONENTIATION ON GALOIS FIELDS FOR CRYPTOGRAPHIC DATA PROTECTION SYSTEMS

Al-Mrayt Ghassan Abdel Jalil Halil, O. Markovskyi, A. Stupak

The article proposes the organization of accelerated execution of the basic operation of a wide range of cryptographic algorithms with a public key-exponentiation on finite Galois fields GF(2^n). Acceleration of the computational implementation of this operation is achieved by organizing the processing of several bits of the code at once during squaring on Galois fields. This organization is based on the use of polynomial squared properties, Montgomery group reduction, and extensive use of previous calculations. Procedures for performing basic operations of exponentiation on Galois fields are developed in detail, the work of which is illustrated by numerical examples. It has been proved that the proposed organization can increase the computational speed of this operation by 2.4 times, which is significant for cryptographic applications.

***Key words**: multiplication operation on Galois fields, cryptographic algorithms based on Galois Fields algebra, Galois Fields exponentiation, Montgomery reduction.*

Introduction

The algebra of finite Galois fields, whose fundamentals were developed in the first half of the 19th century, only gained widespread use in information technology at the beginning of the 21st century. Currently, the mathematical principles of this algebra are the basis for many of the most advanced modern technologies, including mobile communication, high-speed data transmission, mechanisms for restoring lost data, cryptographic data protection, and information security [1]. One of the most significant properties of Galois fields is that regardless of the choice of the generating polynomial, it is feasible to generate a set of algebraic bases whose results will be different [2]. Using this property, it was possible to implement the concept of mathematically distributing communications carried out on the same carrier frequency. The implementation of such a concept in mobile communication systems makes it possible to hold thousands of conversations simultaneously while ensuring their reliable separation. This property is the basis for the application of Galois finite field algebra in modern cryptographic data protection mechanisms. In particular, the algebraic properties of Galois fields are the basis for the implementation of nonlinear transformations in the AES algorithm, which is widely used in practice [3]. A number of protocols for asymmetric encryption, identification, and digital signature with a public key [4], and schemes for cryptographically strong identification of remote users, are based on the Galois field algebra.

It is widely known that the effectiveness of cryptographic data protection mechanisms is determined by the level of security achieved by their use. In addition, it is determined by the speed at which their computation is performed. The last criterion is critical for cryptographic algorithms with a public key, the main computing operation of which is exponentiation performed on huge numbers. When using traditional algebra, this basic operation has the form of modular exponentiation. In Galois field algebra, the result of exponentiation is reduced to the field formed by the fundamental polynomial. The computational complexity of exponentiating n-bit numbers is O(n^3) [5]. This means that with a doubling of the bit depth, the amount of computation increases by a factor of 8. In Galois field algebra, this operation is much faster due to the fact that each bit of numbers is processed independently. In modern conditions, when within the framework of cloud technologies, cybercriminals have remote access to high-powered computer systems, there is an objective need to improve the level of security of cryptographic tools. The only way to enhance protection is to increase the number of bits used. And this dramatically slows down the computational implementation of cryptographic protocols. One of the possible ways out of this situation may be to expand the use of the Galois field algebra and search for ways to speed up the exponentiation of multidigit numbers.

Therefore, the scientific problem of accelerating the computing implementation of the exponentiation operation on Galois fields, which is fundamental to cryptographic applications, is of current relevance to the current stage of development of information and computer technologies.

Problem statement and review of methods for its solution

The expansion of the use of Galois field algebra in modern cryptographic information security protocols, as well as the potential for achieving a higher speed of exponentiation compared to traditional algebra, has led to intensive study of the problem of efficient computational implementation of basic operations in this algebra using hardware and software [6].

When using the Galois field algebra, for each number A=$a_{n-1}\cdot 2^{n-1}+a_{n-2}\cdot 2^{n-2}+\ldots+a_1\cdot 2+a_0$, $\forall j\in\{0, 1, \ldots, n-1\}$: $a_j\in\{0, 1\}$ can be associated with the polynomial A(x)= $a_{n-1}\cdot x^{n-1}+a_{n-2}\cdot x^{n-2}+\ldots+a_1\cdot x+a_0$.

The addition operation on Galois fields is reduced to performing XOR and is further denoted by the symbol '⊕'. Reduction, or finding the remainder from the polynomial division A(x) by the Galois field polynomial P(x), is further denoted as A rem P to distinguish the operation of finding the remainder from dividing the number A by the number M in ordinary algebra: A mod M. Multiplication operation on the Galois fields A⊗B rem P, consists of two operations: polynomial multiplication, denoted by the symbol '⊗', and reduction of the polynomial product with respect to the generating polynomial of the field P. The operation of squaring the number A on the Galois field with the generating polynomial P is denoted as A⊗A rem P or A$|^2$ rem P. Accordingly, the operation of exponentiation on Galois fields, that is, the calculation of the remainder of the polynomial division of the result of raising the number A to the power of E by the polynomial P, is denoted as A$|^E$ rem P.

The existing technologies of exponentiation, both in traditional algebra and on Galois fields, are based on the classical algorithm that provides for the sequential analysis of the bits of the exponent code $E = \{e_{n-1}, e_{n-2}, \ldots, e_0\}$, $\forall j\in\{0,1, \ldots, n-1\}$; $e_i\in\{0, 1\}$. Each step performs a squaring operation on the Galois field and a multiplication operation on the field, depending on the current value of the exponent bit. As each step uses the results of the previous one, the algorithm cannot be parallelized at the bit level of the exponent code.

Currently, there are two versions of this algorithm, which differ in the direction the bits in the exponent code are analyzed. When exponentiating from the high-order digits of the exponent code, at each of n steps, the current result (which is initially equal to one) is squared and multiplied by A if the current bit of the exponent code is equal to one. Correspondingly, the average time t0 of exponentiation from the most significant bits is equal to $1.5\cdot n\cdot t_m$, where t_m is the multiplication time on the Galois fields. As a result of exponentiation from the least significant digits of the exponent, partial parallelization of calculations within a single step is possible. This makes it possible to speed up calculations by a factor of 1.5 [7].

It can be concluded from the above discussion that there is no way to accelerate exponentiation on Galois fields at the level of classical algorithms. This means that speeding up the operation of exponentiation on Galois fields can be achieved by reducing the time of performing the most multiplicative operations on Galois fields: multiplication and squaring [8].

Generally, these operations are divided into two phases: polynomial multiplication (polynomial squaring) and reduction, which involves finding the remainder of the polynomial division of the result of the first phase using the forming polynomial P(x) of the Galois field. The operation of polynomial multiplication of n-bit numbers requires $0.5\cdot n$ logical addition operations and n shift operations and n bit value testing operations to calculate the product. Taking into account that the execution time of the logical addition command is approximately the same as the execution time of the shift command, it can be assumed that the implementation of polynomial multiplication is determined by the execution time of $2.5\cdot n$ logical operations.

During polynomial reduction, the number corresponding to the generating polynomial is added to the current remainder. This operation includes determining the position of the most significant digit of the current remainder, shifting the code of the forming polynomial, logically adding it to the current remainder. Thus, to perform the reduction, it is necessary to perform an average of n bit test operations, $2\cdot n$ shift operations (shifting the code of the generating polynomial and the test code containing one unit), as well as $0.5\cdot n$ logical addition operations. In general, the number of logical operations for performing reduction by dividing polynomials is $3.5 \cdot n$. Thus, the total number of logical operations required to implement the multiplication of n–bit numbers on the Galois fields formed by the polynomial $P(x)$ of degree n is $6 \cdot n$ [8]

The operation of polynomial multiplication is reduced to the logical addition of a maximum of n appropriately shifted multiplicand codes. In theory, the minimum time for this operation is determined by the number of $\log_2 n$ operations of logical addition. Considering the fact that in real applications the value of n is several thousand, the specified approach to accelerating polynomial multiplication can be applied only within the framework of hardware implementations [9].

Almost all researchers consider the reduction operation as the primary source of acceleration for multiplication on Galois fields. This means that further reduction in the time for multiplication is achieved by speeding up the reduction operation. Most of the known methods [10 – 13] are based on the use of previous calculation depending on the constant polynomial P(x), which in cryptographic information protection systems is part of the public key and, accordingly, rarely changes.

In acceleration methods based on the use of this property of the generating polynomial, the remainders from the division of codes 2^{n+1}, …, $2^{2 \cdot n}$ by the generating polynomial $P(x)$ are pre-calculated: $P(x)$: $Q_1 = 2^{n+1}$ rem P, $Q_2 = 2^{n+2}$ rem P, …, $Q_n = 2^{2 \cdot n}$ rem P. The calculated codes are stored in the tabular memory of precalculations. The reduction is reduced to the addition of tabular codes that correlate with the units in the higher n digits of the code of the polynomial product. For this, it is necessary to perform an analysis of the higher n digits of the code of the polynomial product, which requires $2 \cdot n$ logical operations (n operations of testing the value of the bit and n operations of shifting the test code). Another $0.5 \cdot n$ operations are required, on the whole, to add the results of recalculations. Thus, due to the use of previous calculations, it is possible to reduce the average number of logical operations to implement the reduction to $2.5 \cdot n$. At the same time, the total average number of logical operations for multiplication on Galois fields is $5 \cdot n$.

There is another method of speeding up multiplication on Galois fields by combining both phases: polynomial multiplication and reduction using Montgomery technology [14]. In [15], a modification of the Montgomery technology, known in traditional algebra, to the peculiarities of the algebra of Galois fields is proposed. As a result of modifying Montgomery technology for the specifics of Galois fields, the number of logical operations for computing multiplication on Galois fields was reduced to $4.5 \cdot n$.

Purpose and objectives of research

In the current research, the objective is to accelerate the execution of the exponentiation operation on Galois fields, which is essential to the operation of cryptographic protocols. This will be accomplished through the application of precomputation, which facilitates the simultaneous execution of several operations.

In order to accomplish the set goal, the following scientific problems are solved:

– study of the specific properties of the squaring operation on Galois fields, which allow the execution time of several operations to be combined by using the results of previous calculations;

– development, on the basis of the specified specific properties, of the method of accelerated elevation to the square on Galois fields, which, due to the use of previous calculations, allows to combine the operation of adding a multiple and correcting the intermediate result, as well as to combine the processing of several adjacent digits of the multiplier in time;

– analyzing the performance of the developed organization of fast exponentiation on finite Galois fields and comparing it with other known methods designed to accelerate the calculation of exponents;

– study of the proposed organization of fast exponentiation on Galois fields based on software modeling.

Accelerated squaring method on Galois fields with Montgomery group reduction.

The main amount of calculations in exponentiation on Galois fields falls on the operation of squaring. As the main reserves for reducing the number of logical operations when squaring on Galois fields, we can consider:

– use of the property of a polynomial square;
– application of Montgomery reduction modified for Galois fields;
– group processing of discharges when performing the Montgomery reduction.

The basic property of a polynomial square that can be used to speed up calculations is that the polynomial square A⊗A of a binary number A = $a_{n-1}\cdot 2^{n-1} + a_{n-2}\cdot 2^{n-2} +\ldots+ a_2\cdot 2^2 + a_1\cdot 2 + a_0$, де $\forall i\in\{0,1,\ldots,n-1\}$: $a_i \in\{0,1\}$ is equal to the number A⊗A = $a_{n-1}\cdot 2^{2\cdot(n-1)} + a_{n-2}\cdot 2^{2\cdot(n-2)} +\ldots+ a_2\cdot 2^{2\cdot 2} + a_1\cdot 2 + a_0$ [6] This means that polynomial squaring is reduced to inserting zeros between the binary digits of the number A. For example, if A= 14 = 1110_2, then A⊗A = 1010100_2 = 84.

It follows from the above that performing polynomial squaring comes down to shifts in software implementation and permutation of bits in hardware implementation. This means that when using the Montgomery reduction modified for the Galois field, the algorithm for squaring the number A reduces to the following sequence of actions:

1. The cycle counter j is set to zero: j=0, as well as the (n+1)-bit result code R: R=0.
2. Shift R is performed: R>>=1. If the value of j is even, j mod 2 =0, then the most significant digit of r_n is filled with the value of the least significant digit a_0 of the number A: $r_n = a_0$. Shift A: A>>=1. If the value of j is odd, then the most significant bit of r_n is filled with zero: r_n =0. Increment j: $j = j$+1. If $j < n$, return to repeat step 2. If $j > 2\cdot n$ go to step 4
3. If r_0=0, then code P is logically added to the current result P: R = R ⊕ P. Return to repeat step 2.
4. End of procedure. The value R = A⊗A⊗ U^{-1} rem P, U^{-1} is the multiplicative inversion of the polynomial Q(x)=x^n on the Galois field formed by the polynomial P(x), i.e. U⊗U^{-1} rem P =1.

In order to obtain the correct value of the square of the number A on the Galois field, the result of the procedure should be multiplied by U: R′= R⊗U rem P. However, the specified correction is not performed during exposure.

The described procedure of squaring on the Galois field is illustrated by the example of squaring the number A = 12_{10} = 1100_2 on the Galois field, formed by the polynomial $P(x) = x^4+x^2+x+1$, which corresponds to the number P=10111_2=23_{10}; n =4, a U=10000_2 = 32, U^{-1} = 8_{10}=1000_2. Indeed, U·U^{-1} rem P = 32·8 rem 23 = 1. Real result R′=A⊗A rem P = 12⊗12 rem 23 = 12. Step-by-step change of variables R and A in the process of performing the above procedure of squaring A = 12 on the Galois field, with a generating polynomial $P(x) = x^4+x^2+x+1$ is shown in Table 1.

The result R is the product A⊗A⊗U^{-1} rem P = 12⊗12⊗8 rem 23 = 9.To obtain the correct value of the square of the number A=12 on the Galois field, multiply the result R by the value U: R′=R⊗U rem P = 9⊗16 rem 19 = 12.

The execution of the above procedure involves performing n shifts of the number A, $2\cdot n$ shifts of the number R, on average $0.5\cdot n$ logical addition operations (XOR), n bit value testing operations. Thus, the total number of logical operations required to implement the proposed squaring procedure on the Galois field is $3.5\cdot n$.

The main advantage of the proposed procedure is that it eliminates the testing of bits of the multiplier A. This opens up opportunities for group processing of several digits of the number and, thereby, reducing the amount of required calculations.

To theoretically substantiate the possibility of Montgomery group reduction, we prove that for any intermediate result code R = $r_n\cdot 2^n + r_{n-1}\cdot 2^{n-1}+\ldots+r_{k-1}\cdot 2^{k-1}+\ldots+r_1\cdot 2+\mathrm{r}_0$, where $\forall j\in\{0,1,\ldots,n\}$: $r_j\in\{0,1\}$, there is a linear combination L(P) of no more than k shifted codes P: L(P) = $v_{k-1}\cdot 2^{k-1}\cdot P + v_{k-2}\cdot 2^{k-2}\cdot P+\ldots+v_1\cdot 2\cdot P+v_0\cdot P$, $\forall i\in\{0,1,\ldots,k-1\}$: $v_i\in\{0,1\}$, such that their k lower digits are equal to k least significant digits of R. The considered linear combination L(P) of shifted k codes P corresponding to the generating polynomial P(x) of the nth degree of the Galois field can be represented as an (n+k)-bit code D: L(P) = $v_{k-1}\cdot 2^{k-1}\cdot P + v_{k-2}\cdot 2^{k-2}\cdot P+\ldots+v_1\cdot 2\cdot P+v_0\cdot P$ = D = $d_{n+k-1}\cdot 2^{n+k-1} + d_{n+k-2}\cdot 2^{n+k-2}+\ldots+d_1\cdot 2\oplus d_0$

Each i-th bit d_i from among the k least significant bits of the code D can be represented as a logical sum of pairwise products of bit components v_0, v_1, …, v_i and bit values p_0, p_1, …, p_i such that the sum of their indices is equal to i :

$$d_i = v_0 \cdot p_i \oplus v_1 \cdot p_{i-1} \oplus \ldots \oplus v_i \cdot p_0 = \bigoplus_{j=0}^{i} v_j \cdot p_{i-j}\,. \qquad (1)$$

Table 1.

Dynamics of changes in variables R and A when performing the procedure of squaring A=12 on the Galois field formed by the polynomial $P(x) = x^4+x+1$

j	Transformation *R*	Transformation *A*
0	R=0 R>>1 = 00000	A=1100 A>>1 = 0110
1	R= 00000 R>>1 = 00000	A=0110
2	R= 00000 R>>1 = 00000	A=0110 A>>1 = 0011
3	R= 00000 R>>1 = 00000	A=0011
4	R= 10000 R>>1 = 01000	A=0011 A>>1 = 0001
5	R=01000 R>>1 = 00100	A=0001
6	R= 10100 R>>1 = 01010	A=0001 A>>1 = 0000
7	R= 01010 R>>1 = 00101 R⊕P = 10010	A=0000
8	R= 10010 R>>1 = 01001	

If we take into account that the generator polynomial $P(x)$ of the Galois field is prime, then p_0=1.. With this in mind, the expression for the *i* -th digit d_i of the number D can be represented as:

$$d_i = v_i \oplus \bigoplus_{j=0}^{i-1} v_j \cdot p_{i-1-j}\,. \qquad (2)$$

In order to prove that for any of the 2^k–1 possible combinations (except zeros) of values of the *k* least significant digits of the number R, one can find a linear combination L(P) of codes P shifted by no more than *k* digits, it is necessary to show that for any code $r_{k-1},r_{k-2},\ldots,r_1,r_0$ (except zeros) there exists $v_{k-1},v_{k-2},\ldots,v_1,v_0$,such that $\forall i \in \{0,1,\ldots,k-1\}$: $r_i = d_i$.. This condition is satisfied if there is a solution for the following system of linear equations:

$$\begin{cases} r_0 = v_0 \\ r_1 = v_1 \oplus v_0 \cdot p_1 \\ r_2 = v_2 \oplus v_1 \cdot p_1 \oplus v_0 \cdot p_2 \\ \qquad \vdots \\ r_{k-1} = v_{k-1} \oplus v_{k-2} \cdot p_1 \oplus \ldots \oplus v_0 \cdot p_{k-1} \end{cases} . \qquad (3)$$

An analysis of system (3) shows that it has a unique solution. Indeed, the value of v_0 is easily found from the first equation of systems (3): v_0=r_0. The second equation, taking into account the found value v_0=r_0, contains only one unknown component v_1, the value of which is uniquely found in the form: v_1=r_1⊕r_0·p_1. Similarly, the third equation of system (3), taking into account the found valuesv_0 and v_1, contains only one known value v3, which is uniquely in the form: v_3=r_2⊕ p_1·(r_0 ⊕ r_1) ⊕ r_0·p_2. Thus, the analysis of system (3) shows that each of its following equations, including into account the previously identified unknowns, contains only one unknown component, which can be uniquely found from this equation. This means that system (3) always has a unique solution, that is, there always exists a linear combination of numbers P shifted by no more than *k*-1 positions, such that its lower k digits are equal to the lower *k* digits of an arbitrary number R.

By the proved statement, one can perform Montgomery reduction by *k* digits of the current result simultaneously when squaring on Galois fields. This will significantly speed up the basic operation of exponentiation on Galois fields.

To do this, it is proposed once for a given generating polynomial P(x) of the Galois field for each of the possible 2^k–1 (except for zeros) combinations of the *k* -bit code $r_{k-1},r_{k-2},\ldots,r_1,r_0$ to calculate the values of the sums L(P) = $v_{k-1}\cdot 2^{k-1}\cdot$P + $v_{k-2}\cdot 2^{k-2}\cdot$P+…+$v_1\cdot 2\cdot$P+$v_0\cdot$P, in which the values of *k* least significant digits are equal to the above combination. For given values of r_{k-1}, …, r_0, the corresponding values v_{k-1}, v_{k-2}, …, v_1, v_0 are found as a result of solving the system of equations (3). The calculation results are presented in the form of 2^k–1 tabular values T(1), T(2), …, T(2^k–1).

The value of *k* is chosen to be even and such that *n* is evenly divisible by it.

The foregoing is illustrated by the following example. Let n=8 and the Galois field is formed by the polynomial P(x)=x^8+x^4+x^3+x^2+1 For *n*=8, the number U =2^n =256, and its multiplicative inversion U^{-1} with the above generating polynomial P(x) is equal to U^{-1}=127; indeed 256⊗127 rem P(x) =1.

This polynomial corresponds to the number P=100011101_2 = 285_{10}. The lower four digits (for *k*=4) of this number are: p_0=1, p_1=0, p_2=1 и p_3=1 and p_3=1. In order to determine the values of v_0, v_1, v_2 and v_3 at which the lower three digits of the linear combination $v_3\cdot 2^3\cdot$P ⊕ $v_2\cdot 2^2\cdot$P ⊕ $v_1\cdot 2\cdot$P ⊕ $v_0\cdot$P are equal to 1010, i.e. r_3=1, r_2=0, r_1=1, r_0=0it is necessary to solve the system of equations (4) which, in the framework of the example, has the following form:

$$\begin{cases} 0 = v_0 \\ 1 = v_1 \\ 0 = v_2 \oplus v_0 \\ 1 = v_3 \oplus v_1 \oplus v_0 \end{cases} \tag{4}$$

Substituting the found value v_0=0 into the third equation, it is easy to determine v_2=0. Similarly, v_3=0 is deduced from the fourth equation. The found values determine the linear combination: 4·P ⊕ P = 2·285= 570_{10} = 0010 0011 1010_2. Thus, the table value T[1010] = T[10] = 570. The four least significant bits of this linear combination are equal to 1010. Similarly, linear combinations can be constructed for all possible 4-bit codes from 0001 to 1111, the values of which are summarized in Table 2.

In addition, to quickly form *k*-bit fragments of a polynomial square from *k / 2*-bit fragments of a number by inserting zeros between their bits, it is proposed to create and use a Z table. Such a table contains polynomial squares obtained by inserting zeros for each of $2^{k/2}$–1 *k*/2-bit codes. In particular, *k*=4 table Z consists of three rows: Z[1] = Z[01_2] = 0001_2, Z[10_2] = 0100_2 and Z[11_2] = 0101.

The actions outlined above, depending only on the generating polynomial P(x) and the number *k* of simultaneously processed bits, are carried out only once for cryptographic data protection systems, since the polynomial is part of the public key.

Calculation of the square A⊗A rem P of the number A on the Galois field is proposed to be performed in the following sequence:

1. The cycle counter j is set to zero: *j*=1, as well as the (*n*+*k*)-bit result code R: R=0.
2. R is shifted by *k* bits: R>>=k. The upper k digits of R are filled with a table code, the number of which is determined by the lower *k*/2 digits of A: Z($a_{k/2-1},a_{k/2-2},\ldots,a_1,a_0$).
3. If the lower *k* bits of R: $r_{k-1},r_{k-2},\ldots,r_0$ are equal to zero, go to step 4. Otherwise, the code T[$r_{k-1},r_{k-2},\ldots,r_0$] is logically added to R: R = R ⊕ T[$r_{k-1},r_{k-2},\ldots,r_0$].
4. A is shifted by *k*/2 bits: A>>=*k*/2.. Increment *j*: *j*=*j*+1. If j ≤ 2·*n*/*k*, return to repeat step 2.

The following example illustrates the proposed procedure for accelerated squaring on Galois fields. Let it be necessary to square the number A=172_{10} = 1010 1100_2 on the Galois field with the generating polynomial P(x)=x^8+x^6+x^4+x^3 for which table 2 is constructed for k=4. The true value of the result A⊗A rem P = 172 ⊗ 172 rem 285 = 11111_2=31.

The dynamics of changes in R and A over steps j of the described procedure for accelerated squaring on Galois fields is shown in Table 3.

The result R=66 differs from the true one and is the product $A \otimes A \otimes U^{-1}$ rem P = 172⊗172⊗147 rem 285. To obtain the real square R′ of the number A=172 on the Galois field, it is necessary to perform the Montgomery correction, that is, multiply the result R by the value U: R′=R⊗U rem P = 66⊗256 rem 285 = 31.

Table 2.

Tabular values of the results of precomputations for the Galois field with generating polynomial $P(x)= x^8+x^4+x^3+x^2+1$ for k=4

r_3,r_2,r_1,r_0	T	r_3,r_2,r_1,r_0	T
		1 0 0 0 (8)	2280_{10} = 1000 1110 1000_2
0 0 0 1 (1)	1425_{10} = 0101 1001 0001_2	1 0 0 1 (9)	1385_{10} = 0101 0110 1001_2
0 0 1 0 (2)	2850_{10} = 1011 0010 0010_2	1 0 1 0 (10)	570_{10} = 0010 0011 1010_2
0 0 1 1 (3)	1875_{10} = 0111 0101 0011_2	1 0 1 1 (11)	4027_{10} = 1111 1011 1011_2
0 1 0 0 (4)	1140_{10} = 0100 0111 0100_2	1 1 0 0 (12)	3228_{10} = 1100 1001 1100_2
0 1 0 1 (5)	2565_{10} = 1111 0011 0101_2	1 1 0 1 (13)	285_{10} = 0001 0001 1101_2
0 1 1 0 (6)	3990_{10} = 1111 1001 0110_2	1 1 1 0 (14)	1710_{10} = 0110 1010 1110_2
0 1 1 1 (7)	855_{10} = 0011 0101 0111_2	1 1 1 1 (15)	3135_{10} = 1100 0011 1111_2

Table 3

Step by step changes of variables R and A in each step execution of the procedure when squaring A⊗A rem P for A=172 and P=285 for k=4.

j	Transformation R		Transformation A (A>>=2)
	XOR	Shift (R>>=4)	
0	0000 0000	0000 0000 0000	1010 1100
1	–	0000 0000 0000	0010 1011
2	–	0011 0000 0000	0000 1010
3	–	0010 0011 0000	0000 0010
4	R=R⊕T[3] = 547⊕1875 = 0101 0100 0000	0000 0101 0100	0000 0000
5	R=R⊕T[4] = 84⊕1140= 0100 0010 0000	0000 0100 0010	

Analysis of the obtained results

The main advantage of the proposed method of performing the exponentiation operation on Galois fields is to speed up its computer implementation. This makes it possible to accelerate the implementation of a wide range of cryptographic data protection protocols accordingly.

When exponentiation on Galois fields is utilized in information security systems, the real length n (typically 2048 or 4096) of operands is 1 – 2 orders of magnitude greater than the capacity of the processor. Therefore, when estimating the number of operations required for squaring, one can neglect operations on operands whose size is less than the processor capacity and take into account only operations on “long”, that is, n-bit operands.

The execution of the procedure described above includes performing n/k shifts of the number A, $2{\cdot}n/k$ shifts of the number R, n/k operations of logical addition (XOR). Thus, the total number of logical operations required to implement the proposed squaring procedure on the Galois field is $4{\cdot}n/k$. This means that the use of the group Montgomery reduction with processing of k digits at once makes it possible to speed up squaring on Galois fields by $0.75{\cdot}k$ times.

Conclusion

Conducted research aimed at speeding up the computational implementation of the exponentiation operation on Galois fields, which is basic for elliptic cryptography, yielded the following results.A method of accelerated squaring on Galois fields is proposed and studied, which is distinguished by the fact that it uses the algebraic properties of this operation in combination with the application of group reduction, which allows to speed up this operation. The technology of implementation and application of the proposed method is described in detail. Theoretically and experimentally, it has been proven that the method provides acceleration of the square operation by 6 – 8 times, depending on the number of digits in the group. The exposition is illustrated by numerical examples.

The application of the proposed method for the computational implementation of squaring on Galois fields, which takes 2/3 of the calculations of the exponentiation operation on Galois fields, allows to speed up the execution of this basic operation of a wide range of cryptographic algorithms by 2.4 times.

The developed method is oriented for use in information protection systems based on high-speed public key cryptography.

References

[1] J. Nikolajchyk , *Galose Fields code: theory and application*. Ternopil: Terno-Graph, 2012, p. 576.

[2] M. Rahma, V. S. Gluhov, and I. M. Jolubak, “Principles of construction and design of operational nodes for Galois fields used in cryptographic protection of information based on elliptic curves,” in *Cyber-physical systems: multi-level organization and design*, Lviv: Magnolia-2006, 2019, pp. 58–131.

[3] I. A. Kalmykov, E. C. Stepanov, and K. T. Tincherov, “Development of a method of nonlinear information encryption using the exponentiation operation for a finite Galois field ,” *Modern science-intensive technologies*, no. 9, pp. 84–89, 2019.

[4] B. Schneier and W. Diffie, *Applied cryptography: protocols, algorithms, and source code in C*. Indianapolis (Ind.): Wiley, Cop, 2015, p. 784.

[5] O. P. Markovskiy, Z. Leftherios, and V. R. Maksymuk, “Galois Fields Algebra Utilization for Implementation of the Conception of Zero-Knowledge Under Identification and Authentication of Remote Users,” *Èlektron. model*, vol. 39, no. 6, pp. 33–46, Dec. 2017, doi: https://doi.org/10.15407/emodel.39.06.033.

[6] A. A. Moustafa, “Fast exponentiation in Galois fields GF(2m) using precomputations,” *Contemporary engineering sciences*, vol. 7, no. 4, pp. 193–206, Jan. 2014, doi: https://doi.org/10.12988/ces.2014.3955.

[7] O. Markovskyi, O. Rusanova, A. Olievskyi, and V. Cherevik, “Method of acceleration of exponentiation using precalculations,” *Telecommunications and information technologies*, vol. 58, no. 1, pp. 31–39, 2018.

[8] M. K. Rahma and V. Hlukhov, “Computing square roots and solve equations of ECC over galois fields,” presented at the International Youth Science Forum “Litteris Et Artibus”, 2017.

[9] F. N. Castro, L. A. Medina, and L. Brehsner Sepúlveda, “Closed formulas for exponential sums of symmetric polynomials over Galois fields,” *Journal of Algebraic Combinatorics*, vol. 50, no. 1, pp. 73–98, Sep. 2018, doi: https://doi.org/10.1007/s10801-018-0840-4.

[10] E. M. Popovici and P. Fitzpatrick, “Algorithm and architecture for a Galois field multiplicative arithmetic processor,” *IEEE Transactions on Information Theory*, vol. 49, no. 12, pp. 3303–3307, doi: https://doi.org/10.1109/TIT.2003.820026.

[11] H. Wu, M. A. Hasan, I. F. Blake, and S. Gao, “Finite field multiplier using redundant representation,” *IEEE Transactions on Computers*, vol. 51, no. 11, pp. 1306–1316, doi: https://doi.org/10.1109/TC.2002.1047755.

[12] K. G. Samofalov and A. S. Sharshakov, “A method of accelerated implementation of exponentiation on Galois fields in information protection systems,” *Problems of informatization and control*, vol. 2, no. 33, pp. 143–151, 2011.

[13] V. Osadchyy, “The Order of Edwards and Montgomery Curves,” *WSEAS Transactions on Mathematics*, vol. 19, no. 25, pp. 253–264, May 2020, doi: https://doi.org/10.37394/23206.2020.19.25.

[14] O. Markovskyi, V. Maksymuk, O. Kot, and V. Kuts, “The Employment of Montgomery Reduction for Acceleration of Exponent on Galoise Fields Calculation,” in *Proceeding of International Conference on Security, Fault Tolerance, Intelligence*, 2020, pp. 44–49.

[15] S. Elfard, “Justification of Montgomery Modular Reduction,” *Advanced Computing: An International Journal*, vol. 3, no. 11, pp. 93–96, Sep. 2012, doi: https://doi.org/10.5121/acij.2012.3510.

UDC 004. 004.4 (043.2) DOI: https://doi.org/10.20535/2708-4930.3.2022.265418

ORGANIZATION OF PARALLEL EXECUTION OF MODULAR MULTIPLICATION TO SPEED UP THE COMPUTATIONAL IMPLEMENTATION OF PUBLIC-KEY CRYPTOGRAPHY

I. Boiarshyn, O. Markovskyi, B. Ostrovska

*The article theoretically substantiates, investigates and develops a method for parallel execution of the basic operation of public key cryptography-modular multiplication of numbers with high bit count. It is based on a special organization of the division of the components of modular multiplication into independent computational processes. To implement this, it is proposed to use the Montgomery modular reduction. The described solution is illustrated with numerical examples. It has been theoretically and experimentally proven that the proposed approach to parallelization of the arithmetical process of modular multiplication makes it possible to speed up this important for cryptographic tasks operation by 5 – 6 time*s.

Key words: *modular multiplication, Montgomary modular reductions, open key cryptography, parallel computation, multiplicative operations of modular arithmetic*.

Introduction

The process of modular exponentiation, which is executed on numbers whose bit count significantly exceeds the bit count of the processor, is the fundamental operation for a wide range of cryptographic algorithms which are based on irreversible problems of number theory. In particular, this operation is the basis of computational implementation of RSA, El-Gamal, digital signature standard, FESIS scheme of strict identification of remote users [1].

The protection level of cryptographic security mechanisms, which are based on the operation of modular multiplication, is fully determined by the bit count of the module [2]. To date, 2048 bits have been enough for most practical tasks.

At the same time, an analysis of the dynamics of the improvement of applied problems in which public key data protection mechanisms are practically used shows that a significant part of them is performed in real time and requires fast implementation of the corresponding calculations. Another vital feature of the use of modular arithmetic at the present stage of development of public key cryptography is the increase in the number bit count used. The dynamics of the improvement of cloud technologies potentially provides attackers with the ability to remotely access large computing power, which can be used to break cryptographic protection mechanisms. This catalyzed the need for an adequate increase in the level of security, which for cryptographic mechanisms with a public key can be achieved by increasing the bit count. This leads to a noticeable increase in the time of computational implementation of cryptographic data security mechanisms.

Therefore, the task of scientific research is to speed up calculations that implement public key cryptographic mechanisms by using the multiprocessor capabilities of modern computer systems. The main way of solving this problem is the parallelization of the basic operation – the modular multiplication.

The scientific problem of speeding up modular multiplication for cryptographic information security systems is relevant for the present stage of development of information and computer technologies.

Problem statement and review of methods for its solution

Modern public key cryptography was founded at the operation of modular exponentiation. The problem of rapid implementation of this operation is of key importance for the development of information protection complexes and information security.

For personal computers and powerful systems, this problem can be solved by including crypto processors in the hardware. To date, a significant range of cryptoprocessors [4] is commercially produced, almost all of which implement the modular exponentiation operation at the hardware level.

But for a wide class of mobile computer devices, terminal microcontrollers of systems for remote control of real-world objects, in which the Internet is used as a data exchange medium, the problem of fast implementation of the modular exponentiation operation is very acute. For many critical applications, the use of crypto processors is unacceptable for information security reasons.

It is a widespread knowledge that the classical scheme of modular exponentiation is strictly sequential and practically cannot be parallelized [5]. Therefore, the main point for increasing the speed of calculating the modular exponent is the parallelization of its fundamental operation – modular multiplication.

The operation of modular multiplication of numbers of large bit count $A \cdot B \ mod \ X$ consists of two parts: multiplication of the components $Y=A \cdot B$ and finding the residue after dividing the product $A \cdot B$ by the module X. In public key cryptography, the module X is part of the public key, so it can be considered constant [6].

Algorithms for modular multiplication are divided into two groups: gradual, in which the multiplication $A \cdot B$ and the gradual calculation of the residue from division by the module are execute sequentially in time and alternating, in which the operations of multiplication and finding the remainder of the division are combined in time [7].

For modular multiplication algorithms of the first group, there are special possibilities to use the processor's built-in multiplication instructions. To do this, N numbers that take part in the modular multiplication operation are divided into K fragments, the length S of which is equal to the processor bit count. Accordingly, the procedure of modular multiplication is reduced to pairwise multiplication of fragments with gradual summation of the results obtained. This method makes it possible to use the hardware of modern processors with high efficiency, and, in particular, the built-in circuits for multiplication acceleration [8].

In the basic algorithm, modular reduction is executed using the operation of integer division of a 2·s-bit dividend by a s-bit divisor to obtain a quotient and a remainder [9]. Since the division of N-bit numbers on a s-bit processor (N>>s) is very inefficient, the reduction in the basic algorithm requires $k\cdot(k+2.5)$ multiplication operations and k integer division operations [10]. To date, a number of algorithms [11, 12] have been declared to improve the performance of the software implementation of the modular multiplication operation. Most of them implement an increase in the performance of modular multiplication due to the acceleration of modular reduction by eliminating the operation of integer division, which is used in the basic algorithm [9].

Two technologies for finding the modulo modulo remainder have received the widest practical use: Barrett's algorithm [9] and Montgomery algorithm [13]. The first one is funded in calculating the minimum value of m for which $A\cdot B - m\cdot X < X$. Accordingly, the remainder of the division is calculated as $R = A\cdot B - m\cdot X$. Thus, Barrett's algorithm is implemented with two s-bit multiplications, while modular multiplication $A\cdot B \ mod \ X$ using Barrett's algorithm is implemented with three.

Another technology for calculating the modulo product residue, the Montgomery algorithm, is well adapted to the universal processor architecture. The algorithm replaces the operation of division by a random modulus X with divisions by a power of 2, which are effectively implemented by shifts. The modular reduction operation in Montgomery's algorithm requires $k\cdot(k+1)$ multiplication operations.

The total computational complexity of the implementation of the algorithm for modular multiplication of N-bit numbers using the Montgomery algorithm on a s-bit processor is determined by $2\cdot k2$ + s processor multiplication operations and $4\cdot k2+4\cdot k+2$ processor addition operations. *A* significant advantage of the Montgomery algorithm is that it is relatively easy to combine in time with the multiplication process. This allows, in the process of modular multiplication, to limit the length of intermediate results to $N + 1$ and thereby reduce the amount of calculations compared to the sequential scheme that works with 2·N-bit intermediate results [14].

Two approaches are most often used [15] to speed up modular multiplication:

– precalculations depending only on the value of the module, which are stored in a special table memory;

– simultaneous processing of several digits of the multiplier;

– parallelization of operations of summation of fragments of a modular product.

In practice, these three approaches are often used in combination.

The analysis of existing methods to speed up the execution of modular multiplication showed that with limits of computational processes that run on a single processor, the possibilities of obtaining new results are practically exhausted. This means that a further increase in the speed of the computational implementation of the modular multiplication operation, necessary for practical problems, can be achieved only by using the capabilities of multi-core processor architectures.

Purpose and objectives of research

The target of the research is to speed up the execution of the modular multiplication operation on numbers, which is important for cryptographic tasks, the bit count of which significantly exceeds the bit count of the processor, due to the organization of parallel calculation of fragments of the modular product on multi-core computers.

The following set of tasks is solved in the work to achieve the target goal:

– analysis of the computing process of modular multiplication due the point of view of its parallelization possibilities; description for choosing a scheme of modular reduction, which combining in time with the multiplication process;

– creating of a method of parallel modular multiplication using a multi–core architecture, which, due to the division of the computing process into loosely connected fragments, allows to organize their parallel processing, due to which acceleration of the computational implementation of modular multiplication is achieved;

– optimization of the structure of the parallel computation of the Modular multiplication according to the criterion of maximum exploitation of processor elements;

– theoretical evaluation of the effectiveness of the developed method of accelerated modular multiplication;

– software development and experimental evaluation of the effectiveness of the proposed method of parallel modular multiplication of numbers, the bit count of which significantly exceeds the bit count of the processor.

The object of research to which the article is devoted are the processes of calculating multiplicative operations of modular arithmetic, which are performed on numbers, the length of which is orders of magnitude greater than the bit capacity of processors.

The method of parallel calculation of the modular product on multicore processors

To achieve this target, is declared the following organization of parallel computation of the modular product $A \cdot B$ mod X in the form of s independent computational processes. Accordingly, these computational processes are performed on s cores. For this, the N-bit factor A=a1+a1·21+a2·22+…+aN·2N, $\forall j\in\{1, 2, \ldots, N\}$: $aj\in\{0,1\}$, is decomposed into s partial factors A1, A2, …, As with N bits. Each i-th, $i \in\{1,2, \ldots, s\}$, N-bit partial multiplier Ai includes those r=N/s digits of the multiplier A, the residue of dividing by s numbers of which is equal to i, the remaining digits of the N-bit partial multiplier multipliers are zero. In other words, each i–th partial factor Ai can be represented as:

$$A_i = \sum_{j=i}^{(r-1)\cdot s} a_j \cdot 2^j \qquad (1)$$

The above can be illustrated by the following example: if the bit depth is N=12 and the number of independent computing processes is s=3, then the factor A = $0011\ 1001\ 1100_2 = 924_{10}$, is divided into 3 partial factors, each of which contains r=N/s=4 significant binary bits of the full multiplier A. The partial multiplier A_0 includes every fourth, starting from the least significant, that is, the 1st, 4th and 7th bits of the full multiplier A: A_1=$0010\ 0000\ 1000_2$. The second partial multiplier A_1 contains the 2-nd, 5-th and 8-th digits of the full factor A: A_2=$0000\ 1001\ 0000_2$.The last, third partial factor is: $A_3 = 0001\ 0100\ 0100_2$.

It is quite obvious that the disjunction of all partial factors is equal to the factor A of the modular product: $A_1\cup A_2 \cup\ldots \cup A_s = A$, and the conjunction of partial products is equal to zero: $A_1\cap A_2 \cap\ldots \cap A_s = 0$. This means that the modular product $A\cdot B$ mod X is equal to the sum of modular products of partial factors and multiplier B:

$$A \cdot B \bmod m = (\sum_{i=1}^{s} A_i \cdot B \bmod m) \bmod m \tag{2}$$

Therefore, the separation of the significant digits of the factor A by partial factors $A_1, A_2, \ldots, A_s$, which are independently modularly multiplied by the multiplier B, provides an increase in the speed of calculating the modular product due to the parallelization of the process of modular multiplication. In addition, the predominance of zeros in each of the partial factors, which are independently multiplied by the multiplicand, creates conditions for the effective use of precalculations in the process of calculating modulo modulus residues using the Montgomery technology [2].

Montgomery's algorithm is funded on the idea of replacing the calculation of modular reduction modulo X with the calculation of reduction modulo M, which is a power of 2, so that division operations are reduced to shifts.

Montgomery's technology allows instead of calculating Y mod X to calculate $R=Y \cdot M^{-1}$ mod X without the division operation, where M^{-1} is the modular inversion of M. After that, to obtain Y mod X the calculated value of R is multiplied by M mod X: Y mod X = ($Y \cdot M^{-1}$ mod $X \cdot$ M mod X) mod X = ($Y \cdot M \cdot M^{-1}$) mod X = ($Y \cdot 1$) mod X. Thus, to compute Y mod X one must compute M mod X. However, in practice, applying the Montgomery reduction $X < M < 2 \cdot X$, so that M mod $X = M–X$. That is, the calculation of M mod X is reduced to one operation of subtracting N-bit numbers. Usually $M=2^N$, and the module X is a number of length N binary digits, and the most significant bit of the binary representation of X equal to one: $X_{N-1}=2^{N-1}$.

When calculating $A \cdot B$ mod X, the complexity of pre-computation and post-computation must be taken into account. If we take into account the complexity of calculating the modular product, taking into account the correction, which again requires the operation of modular multiplication by the modular inversion of 2^N modulo X, and multiplying it by the result using the Montgomery algorithm, then it turns out that the complexity will be $4 \cdot X \cdot (X+1$**)**. This means that when performing a single operation of modular multiplication, the Montgomery algorithm has no obvious advantages over the basic algorithm.

An analysis of these features allows us to formulate requirements for the organization scheme of partial calculations. The implementation of this computation, together with the preservation of the general principles on which the Montgomery algorithm is based (for example, minimization of all intermediate results by replacing them with smaller numbers congruent in a given modulo), allows us to obtain a working algorithm of parallel multiplication, which is more efficient. In this case, the organization of calculations must be single-pass: to obtain the result of modular multiplication, the calculation cycle must be performed only once. The classical Montgomery algorithm does not satisfy this requirement, because it is two-way. To form a general result based on the results of partial multiplications, the maximum load of processor cores is required.

As pointed, a characteristic feature of the declared variant of dividing the factor A into partial factors is the presence of local groups of zeros, the number of which is not less than s. This allows us to solve the problem of accelerating the calculation of the modular product by adding to the intermediate result not a module, but a linear combination of the module P(X) chosen in such a way that the lower s digits of the sum of the intermediate result and this linear combination $Y+P(X)$ are equal to zero. Accordingly, after that, the resulting sum is shifted to the right by s bits at once without loss of significant bits. It is quite obvious that such a solution makes it possible to speed up the reduction of the intermediate result by a factor of s at once. For the practical implementation of the proposed technology of accelerated calculation of the remainder of the division of the intermediate result by the module X it seems necessary to calculate in advance for each of the 2^s options for possible values of the lower s digits of the intermediate result Y the value of the linear combination of the module $P(X)$, the lower s digits of which are the algebraic complement of s lower digits of the intermediate result. The results obtained from such pre-calculations are stored in the form of a table T of pre-calculations. An example of the table T precomputation P(X) is given below in Table 1 for s=3 and the value of the module X= 23⊕29 = 667.

The full capacity of table memory for storing the results of precomputations P(X) is $N \oplus 2^s$ bits. he way of storing P(X) in one table presented above can be considered as a special case of partitioned organization of tables of precomputation results. The use of multi-section tables can significantly reduce the amount of memory for their storage.

For example, under the conditions of the above example of the implementation of accelerated multiplication of 2048-bit numbers on an 8-bit microcontroller with two-section memory, the amount of memory required will be 8.57 times less than with a single-section organization of table memory. On the other hand, the use of a multi-section organization of table memory is associated with an increase in the execution time of modular reduction.

Table 1.
Linear Combination Precomputation Example $P(m)$
for module X= 23·29 = 667 and S=3

low-order Y	$P(X)$	low-order P(X)
$y_3\, y_2\, y_1$		$p_3\, p_2\, p_1$
0 0 1	3335 = 4⊕X + X	1 1 1
0 1 0	1334 = 2⊕X	1 1 0
0 1 1	4669 = 4⊕X + 2⊕X + X	1 0 1
1 0 0	2668 = 4⊕X	1 0 0
1 0 1	667 = X	0 1 1
1 1 0	4002 = 4⊕X + 2⊕X	0 1 0
1 1 1	2001 = 2⊕X + X	0 0 1

The presented method of modular multiplication with parallelization of calculations on s processor cores involves the simultaneous execution of procedures for calculating partial products on all processor cores, followed by their cascaded modular summation to reduce the time of generating the result of modular multiplication.

In this case, the procedure for calculating a partial modular partial product consists in performing the following sequence of actions:

1. The counter h of cycles is set to zero, as well as the Y code of the current result h=0; Y=0.
2. The partial factor A_r with the number r is shifted to the right by r–1 binary digits: A_r=A_r>>(r–1) with the high digits filled with zeros.
3. If the least significant digit of the partial product Y is equal to one: y_1 =1, then the multiplier B is added to the result code: Y += B.
4. If the value of the counter h of cycles is a multiple of the value s, then go to step 6.
5. To the code of the partial product Y the tabular code T[l], is added, addressed by s least significant digits of the result code $l = y_1 + 2 \oplus y_2 + \ldots + 2^{s-1} \oplus y_s$: Y += T[l].
6. Result code Y and multiplier B are shifted s bits to the right: Y= Y>>s; B = B>>s. The cycle counter h of the algorithm is increased by one: h++, the transition to the repeated execution of paragraph 3 of the algorithm is performed.
7. To the code of the partial product Y, the tabular code T[l] is added, addressed by s–r lower significant digits of the result code$l = y_1 + 2 \oplus y_2 + \ldots + 2^{s-r-1} \oplus y_{s-r+1}$: Y += T[l].
8. The Y result code is shifted s–r bits to the right: Y= Y >>s–r.
9. End.

After performing the described procedure, Yhas generated a modular product code containing $A \cdot B \cdot M^{-1}$ mod X, where M^{-1} is the multiplicative inversion of M=2^N modulo X. To obtain the correct result, the resulting Y code must be modularly multiplied by M: Y' = $M \cdot Y$ mod X. However, when performing the operation of modular multiplication as a component of modular exponentiation, corrective multiplication by code M is performed only once, after all cycles of the classical algorithm of modular exponentiation have been executed.

Evaluation of the effectiveness of the method of parallel modular multiplication

It is expedient to estimate the efficiency of the proposed method of modular multiplication by means of the achieved acceleration of the computational implementation of this operation when using s processor cores. The numerical expression for the acceleration estimate can be the coefficient q, which is determined by the ratio of the time t_1 for performing modular multiplication in the form of a single process using the Montgomery reduction to the time t for performing this operation in the form of s parallel processes using the developed method:

$$q = \frac{t_1}{t_k} \tag{3}$$

The time t_1 of performing the operation of modular multiplication on one processor using the alternation of the multiplication cycle and the reduction of the precalculation result is determined by the execution time of n cycles by the number of digits of the numbers. Each cycle, depending on the value of the current digit of the multiplier A в, the addition of the multiplier B to the code of the precalculation result Y is performed or not performed. After that, depending on the value of the least significant digit of the received sum Y, the addition of the module X to the code of the precalculation result Y is performed or not performed. ends with shifting the precalculation result code to the right by one bit. Thus, the cycle, on average, contains two operations on n-bit numbers. If the execution time of these operations is denoted by t_N, then t_1=2·N·t_N.

The developed procedure provides for the implementation of the multiplication of the multiplier by the partial factor in the form of N/s cycles, in each of which the following is performed: adding the multiplicand to the result if the least significant bit of the partial factor is equal to one, adding the code from the precalculation table to the result, as well as shifting the multiplier and the result to the right. Accordingly, the average number of operations on n-digit numbers is 3.5, and the value t_X=3.5·N·t_N/s. Thus, the numerical value of the acceleration coefficient q is determined by the following formula:

$$q = \frac{t_1}{t_k} = \frac{2 \cdot N \cdot t_N}{3.5 \cdot t_N \cdot \frac{N}{s}} \approx 0.57 \cdot s \tag{4}$$

Experiments on multi-core processors using a specially developed program showed the role of the acceleration factor equal to 0.5 ·s, close to the predicted theoretical estimate.

Conclusion

As a result of the research aimed at increasing the speed of computer implementation of modular multiplication – the basic operation of public key cryptography based on unsolvable mathematical problems of number theory, the following results were obtained:

Theoretically substantiated, developed and investigated a method for parallelizing the operation of modular multiplication in the form of s independent parallel processes that can be executed on the cores of modern processors, a distinctive feature of which is the division of significant digits of the multiplier into different processes, due to which parallelization is ensured, which allows achieving real acceleration performing this important operation for cryptographic applications. The processing of insignificant digits of partial factors is performed in the form of Montgomery group reduction, which is an additional acceleration factor. To implement group reduction, precomputation tables are used, which depend only on the module and practically do not change, since the module is part of the public key of cryptosystems.

It has been theoretically and experimentally proven that the presented method makes it possible to speed up the computational implementation of modular multiplication by 0.57 · s times.

The developed method is focused on application in multi-core computer systems to accelerate the implementation of a wide range of cryptographic data protection protocols with a public key.

References

[1] L. A. Kabir, O. V. Rusanova, and I. O. Humenyuk, “The method of accelerating modular multiplication according to Montgomery technology,” *Almanac of Science*, vol. 1, no. 52, pp. 44–46, 2022.

[2] K. E. Tribunska, “The method of accelerating modular multiplication using group reduction,” in *Current issues of the development of science and education: materials of the M International Scientific and Practical Conference in Lviv*, Lviv Scientific Forum, Mar. 2022, pp. 38–46.

[3] P. Giorgi, L. Imbert, and T. Izard, “Parallel Modular Multiplication on Multicore Processors,” in *2013 IEEE 21st Symposium on Computer Arithmetic*, pp. 135–142. doi: https://doi.org/10.1109/ARITH.2013.20.

[4] I. Boyarshin, B. Ostrovska, and O. Markovskyi, “Method of accelerated modular multiplication with Montgomery group reduction,” in *Proceeding of International Conference Security, Fault Tolerance, Intelligence*, Kyiv, pp. 40–45.

[5] B. Buhrow, B. Gilbert, and C. Haider, “Parallel modular multiplication using 512-bit advanced vector instructions,” *Journal of Cryptographic Engineering*, vol. 12, no. 1, pp. 46–53, Feb. 2021, doi: https://doi.org/10.1007/s13389-021-00256-9.

[6] V. Osadchyy, “The Order of Edwards and Montgomery Curves,” *WSEAS TRANSACTIONS ON MATHEMATICS*, vol. 19, no. 25, pp. 253–264, May 2020, doi: https://doi.org/10.37394/23206.2020.19.25.

[7] G. Haches , “Montgomery multiplication with no final subtraction,” in *Cryptographic Hardware and Embedded System – CHES’2000.* , 2000, pp. 293–301.

[8] O. P. Markovskyi, O. V. Rusanova, A. A. Olievskyi, and V. M. Cherevyk, “The method of accelerating exponentiation using recalculations ,” *Telecommunication and information technologies*, vol. 58, no. 1, pp. 31–39, 2018.

[9] S. Sherif Elfard, “Justification of Montgomery Modular Reduction,” *Advanced Computing: An International Journal*, vol. 3, no. 5, pp. 41–45, Sep. 2012, doi: https://doi.org/10.5121/acij.2012.3510.

[10] A. V. Anisimov, “Fast direct calculation of modular reduction,” *Cybernetics and system analysis*, no. 4, pp. 3–12, 1999.

[11] S. Kawamura, K. Takabayashi, and Atsushi Shimbo, “A fast modular exponentiation algorithm,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 74, no. 8, pp. 2136–2142, Aug. 1991.

[12] K. G. Samofalov and H. M. Lutskyi, “Effective realization of multiplicative operations of modular arithmetic in information protection systems,” in *Proceeding of International scientific conference*, pp. 435–437.

[13] C. W. Chion and T. C. Yang, “Parallel modular multiplication with table look-up,” *International Journal of Computer Mathematics*, vol. 69, no. 1–2, pp. 22–23, 1998.

[14] A. V. 14. Anisimov, “Algorithmic theory of large numbers,” *Akademperiodika*, p. 153, 2001.

[15] T. Blum and C. Paar, “High-radix Montgomery modular exponentiation on reconfigurable hardware,” *IEEE Transactions on Computers*, vol. 50, no. 7, pp. 759–764, Jul. 2001, doi: https://doi.org/10.1109/12.936241.

UDC 004 *DOI: https://doi.org/10.20535/2708-4930.3.2022.267665*

SIMULATION OF FLUID MOTION IN COMPLEX CLOSED SURFACES USING A LATTICE BOLTZMANN MODEL

V. Kuzmych, M. Novotarskyi

CFD (computational fluid dynamics) modeling is used to determine the distribution of pressure, velocity, and other movement parameters of liquids or gases. Simulation of a fluid flow in a complex closed surface has become demandable in many scientific, medical, and industrial areas. The lattice Boltzmann model is an efficient numerical scheme for modeling fluid flows. In this paper, we investigate nonstationary hydrodynamic processes in closed surfaces using the Boltzmann lattice model to simulate fluid flow in the human stomach.

Keywords: *hydrodynamics, lattice Boltzmann model, simulation*

Introduction

Simulations are widely used in several advanced engineering studies. A suitable numerical method is crucial to obtain accurate results in fields such as fluid flow, thermal transfer, or mechanical engineering.

Today, it is necessary to use an adapted numerical method in complex systems and fields that would be too expensive, dangerous, difficult, or even impossible to study by direct experimentation. Gastrointestinal surgery is a field of study where natural experiments or measurement of various properties of objects or treatment is costly, complex, and unsafe in some cases.

Computational fluid dynamics (CFD) is a branch of hydromechanics that uses numerical analysis and data structures to analyze and solve problems related to the movement of fluids. Computers are used to perform the calculations required to simulate the free flow of a fluid and the interaction of a fluid (liquids and gases) with surfaces defined by boundary conditions. Large and complex problems can be solved with the use of high-speed supercomputers. Modern software provides the accuracy and speed of modeling complex scenarios with transonic or turbulent flows. The first experimental verification of such software is carried out using a wind tunnel, and the final confirmation is carried out during full-scale tests, for example, flight tests.

The gastrointestinal tract is a system such that the health of the whole organism depends on its state. It is known that a disturbance in the balance of proteins, fats, carbohydrates, vitamins, and microelements causes many diseases. All those substances enter into an organism with food. But even the most helpful food products do not become a source of health if the functioning of the gastroenteric tract is violated. In this case, the necessary substances are not assimilated. Therefore, it is essential to pay significant attention to the support of proper functioning of the gastroenteric tract. This problem acquires a particular meaning when dealing with diseases requiring surgical intervention. Among such conditions, the critical place is occupied by the oncologic diseases of parts of the gastroenteric tract, gunshot injuries of the peritoneal cavity, and other illnesses which require reconstruction-recovery operations.

Reconstructive surgery on the human digestive tract can cause negative consequences. These effects were manifested in the appearance of unwanted deformations, so-called "blind bags," which arose due to the formation of zones of high pressure after changes in the geometry of hollow objects of the digestive tract during reconstructive surgery. For this reason, developing a mathematical fluid flow model on the closed surface has become crucial in recent years.

Literature review

We developed the first series of in vitro systems to analyze human digestion [1, 2] at the beginning of the 1990s. Despite the sizeable amount of human and animal digestive tract data, conflicting results have been obtained [3]. The main limitation of this method is the difficulty of reproducing the geometry and motility of the digestive tract. Unfortunately, developing an in vitro system capable of accurately producing the fluid mechanical forces that promote digestion is complicated.

Singh et al. presented an advanced fluid dynamics program that offers a promising technique to characterize the mechanisms promoting digestion [4]. It is possible to use computational fluid dynamics for numerical simulation of the flow of gastrointestinal contents during digestion using knowledge of the motor response of the digestive tract and the physicochemical properties of luminal contents. Pal et al. initially attempted to simulate the gastric flow during digestion [5, 6]. Still, the computational effort required to reproduce the geometry and motility of the stomach prevented an excellent characterization of the system.

Such parameters play the reconstruction-recovery operations, a significant role in the pressure distribution and the field of velocities in the region under study. The mathematical models describing the motion of fluid under the action of peristaltic oscillations are represented most frequently by a system of equations that includes the Navier–Stokes equation and the equation of continuity of a flow [12]. Such an approach is sometimes called “top-down” technology. In this case, fundamental properties of the fluid are used to calculate specific physical parameters. The boundary-value problems, which are formed based on such a system of equations, require significant expenditures of computer time and computational resources for their solution.

Today, modeling fluid flow in a volume with closed-moving surfaces, such as the human digestive tract, requires significant computing resources. Using a probabilistic approach will reduce costs for determining the fluid velocity field. Therefore, this article investigates the possibility of using the Lattice Boltzmann Method (LBM) in fluid flow simulation inside biological objects.

LBM is one of the currently popular methods of computational fluid dynamics, which has been successfully applied to fluid flows through porous media [13], multiphase fluid flows [14], non-Newtonian particle flows [15], and even medical technology [16]. This method differs from traditional CFD methods, such as the finite element method (FEM) and the finite volume method (FVM), which aim to solve boundary value problems based on the Navier-Stokes equation numerically. The mentioned boundary value problem considers a continuous fluid flow. The fundamental difference of LBM is that, in this case, the fluid flow is considered as the movement of particles with elementary fluid volumes. These particles collide in the moving process, changing the parameters of the velocity vector under physical laws.

Our work is devoted to the application of LBM for the simulation of fluid flow processes on complex closed surfaces based on the experience obtained by comparing the properties of fluid flow in different closed complex geometries.

Methodology

The lattice Boltzmann method is a numerical method to solve the Boltzmann equation on a discrete lattice:

$$v \cdot \nabla_x f + F \cdot \nabla_p f + \frac{\partial f}{\partial t} = \hat{\Omega}(f), \tag{1}$$

where F – an external body force, ∇_x, ∇_p, is the gradient in position and momentum space, and $\hat{\Omega}(f)$ is the collision operator. The Boltzmann equation describes the dynamics of a fluid from a microscopic point of view: particles, each with velocities v_i, collide with a certain probability and exchange momentum among each other. For ideal collisions, total momentum and energy are conserved in the collisions. The Boltzmann equation expresses how the probability $f(x, v, t)$ of finding a particle with velocity v at a position x and at time t evolves with time.

Assuming $F = 0$, equation (1) will be next:

$$v \cdot \nabla_x f + \frac{\partial f}{\partial t} = \hat{\Omega}(f). \tag{2}$$

For the sake of simplicity, the collision operator is taken in the most frequently used form:

$$\hat{\Omega}(f) = \frac{1}{\tau}\left(f - f^{(eq)}\right). \tag{3}$$

In (3), τ is a constant defining the time scale, which is necessary for the establishment of local equilibrium, and $f^{(eq)}$ is the density distribution function (so-called Maxwell—Boltzmann distribution function).

Thus, we get the Bhatnagar-Gross-Krook-model (or BGK-model) [7]:

$$v \cdot \nabla_x f + \frac{\partial f}{\partial t} = \frac{1}{\tau}\left(f - f^{(eq)}\right). \tag{4}$$

We make discretization of this model in the space of velocities on a finite set of vectors $\{v_k\}$ with regard for the conservation laws [8]. As a result, we get the system composed of Q equations:

$$\frac{\partial f_k}{\partial t} + v_k \nabla f_k = \frac{1}{\tau}\left(f_k - f_k^{(eq)}\right),\ k = 0,1,2,\ldots,Q-1, \tag{5}$$

where $f_k(x,t) = f(x,v_k,t)$ is the density distribution function associated with the direction of a velocity vector v_k, $f_k^{(eq)}$ is the equilibrium density distribution function corresponding to the vector v_k.

We executed the full discretization of (5) with a time step of Δt and a spatial step of $\Delta x_k = v_k \Delta t$ [13], in order to simplify computer realization:

$$\frac{f_k(x_k + v_k\Delta t, t + \Delta t) - f_k(x_k + v_k\Delta t, t)}{\Delta t} +$$
$$+\frac{f_k(x_k + v_k\Delta t, t) - f_k(x_k, t)}{\Delta x_k} = \frac{-f_k(x_k,t) - f_k^{(eq)}(x_k,t)}{\tau}.$$

Setting $\Delta x_k = \Delta t = 1$, we get the Boltzmann lattice equation

$$f_k(x_k + v_k\Delta t, t + \Delta t) - f_k(x_k,t) = \frac{-1}{\tau}\left(f_k(x_k,t) - f_k^{(eq)}(x_k,t)\right), \tag{6}$$

where x_k is a point in the discretized physical space.

According to the BGK-model, Eq. (6) can be solved with the use of two steps.

1. Collision-related step:

$$\tilde{f}_k(x_k, t + \Delta t) = f_k(x_k,t) - \frac{1}{\tau}\left(f_k(x_k,t) - f_k^{(eq)}(x_k,t)\right) \tag{7}$$

2. Flow-related step:

$$f_k(x_k + v_k\Delta t, t + \Delta t) = \tilde{f}_k(x_k, t + \Delta t) \tag{8}$$

In (7) and (8), the distribution function $\tilde{f}_k$ describes a post-collisional state of the elementary volume of a fluid or the particle of a substance at the point of the discrete space x_k. In the BGK model, the collisions are considered as oscillations of elementary volumes of a fluid relative to the positions of local equilibrium.

The values of elements of the set $\{v_k\}$ are determined in view of the dimension of a model and the number of connected nodes forming the lattice basic element.

The mesoscopic and macroscopic levels of the modeling are connected by means of the following formulas:

$$\rho = \int_{-\infty}^{\infty} f(x,v,t)dv = \sum_{k=0}^{Q} f_i = \sum_{k=0}^{Q} f_k^{(eq)} \tag{9}$$

$$u = \frac{1}{\rho}\int_{-\infty}^{\infty} v \cdot f(x,v,t)dv = \frac{1}{\rho}\sum_{k=0}^{Q} v_k f_k \frac{1}{\rho}\sum_{k=0}^{Q} v_k f_k^{(eq)} \tag{10}$$

where u is the velocity vector of a flow in the fluid, and ρ is the mass density of a flow in the fluid.

Experiments

In order to achieve the practical significance of analysis of fluid flow properties in complex closed surfaces, we prepared two 3D models of the stomach in two different states – a normal state and an anastomosis state. We used Blender software [11] to construct these models. They are displayed in Fig 1, Fig 2.

To apply LBM we discretized each model into a square mesh with the size of $120 \times 78 \times 142$. Parameters of LBM itself are the following: $\Re = 1000$, $\rho = 1000$. We introduced boundary value in the top as a constant flow directed to the bottom, with a velocity equal to 0.1 m/s.

We choose the D3Q19 lattice scheme [10] due to its faster performance in comparison to larger schemes while maintaining acceptable accuracy. The velocity scheme with all v_k vectors is displayed in Fig. 3. This cubic lattice D3Q19 is defined by the following velocities:

$$v_0 = (0, 0, 0)$$
$$v_{1,2}, v_{3,4}, v_{5,6} = (\pm 1, 0, 0), (0, \pm 1, 0), (0, 0, \pm 1)$$

$$v_{7,\dots,10} = (\pm 1, \pm 1, 0)$$
$$v_{11,\dots,14} = (\pm 1, 0, \pm 1)$$
$$v_{15,\dots,18}, = (0, \pm 1, \pm 1)$$

All experiments were performed on a PC with Ryzen 7 5800X CPU and 32 GB RAM, using the Pylbm python library [9].

We measured the magnitude of fluid velocity field distribution at modeling times t = 2.5 sec and t = 5.0 sec. Fig.4–5 show the distribution for the normal state, fig. 6–7 shows the anastomosis state of the human stomach.

Results

Results demonstrated higher velocity magnitude near the bottom part of the stomach in case of anastomosis than in the normal state. In addition, the anastomosis model shows the increased fluid velocity in the “blind bag” under the stomach. In real situations, it can cause the development of negative consequences.

We investigated the relationship between average velocity inside the stomach area and modeling time in the states mentioned above. Fig.8 shows this relationship. During all modeling periods, the average velocity in the normal state is higher than in anastomosis. Due to this outcome and previously mentioned results, we can conclude that the velocity field in the anastomosis state is irregular in comparison to the normal state of the stomach.

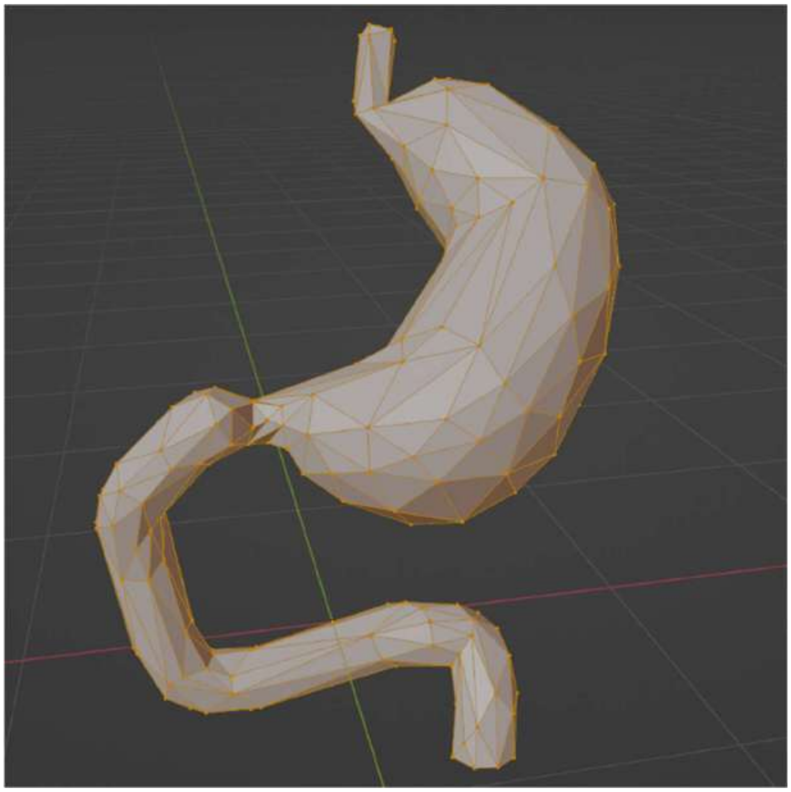


Fig.1. Normal stomach

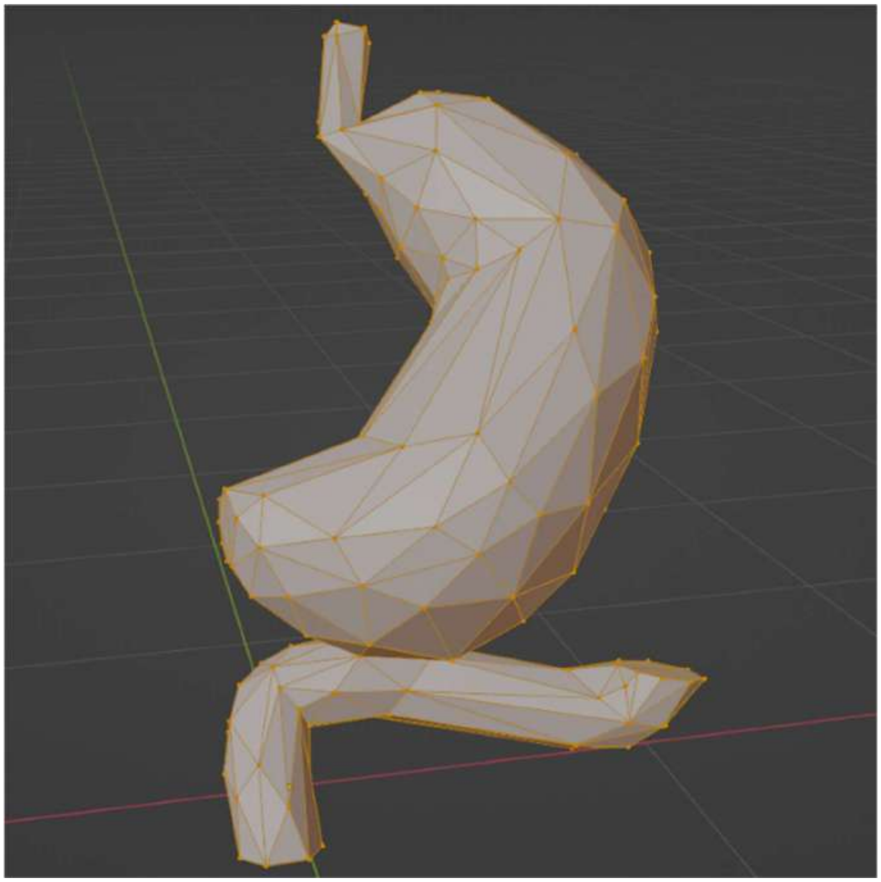


Fig.2. Anastomosis state of the stomach

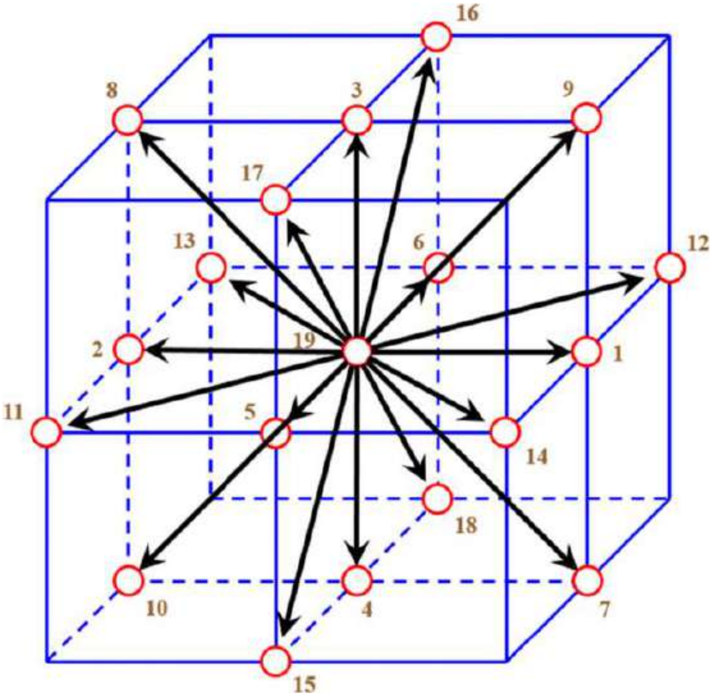


Fig.3. D3Q19 scheme

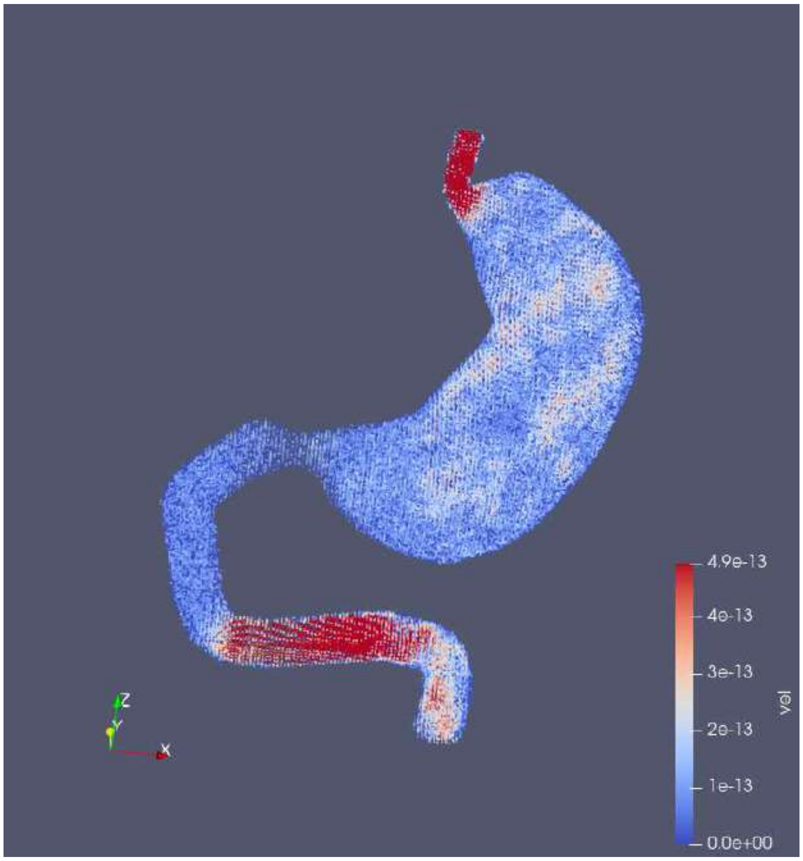


Fig 4. Velocity field distribution in a normal state at time 2.5 sec

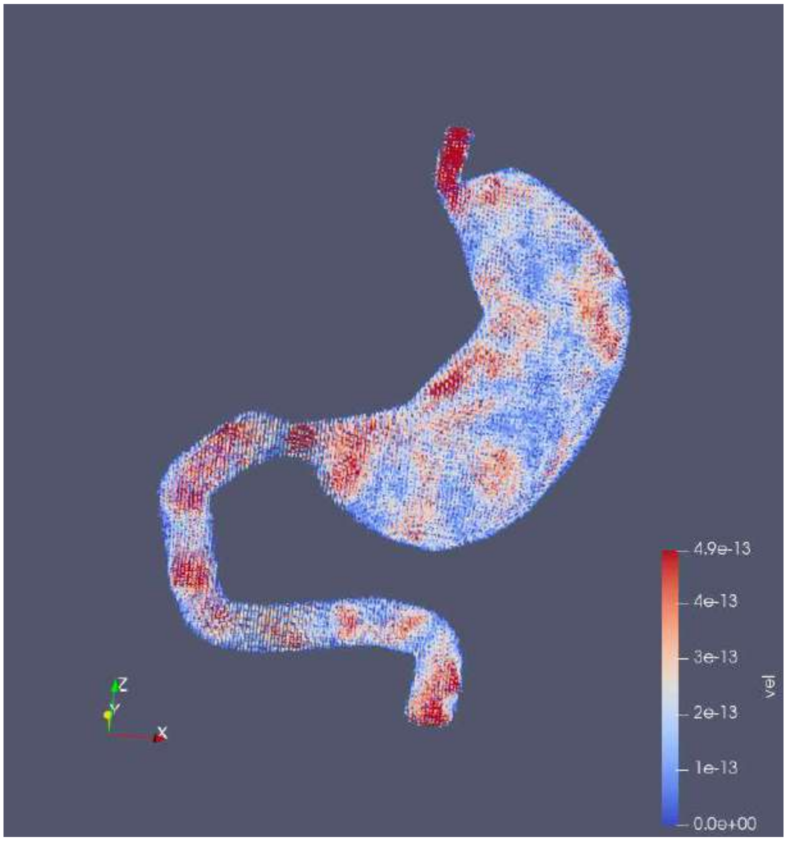


Fig 5. Velocity field distribution in a normal state at time 5,0 sec

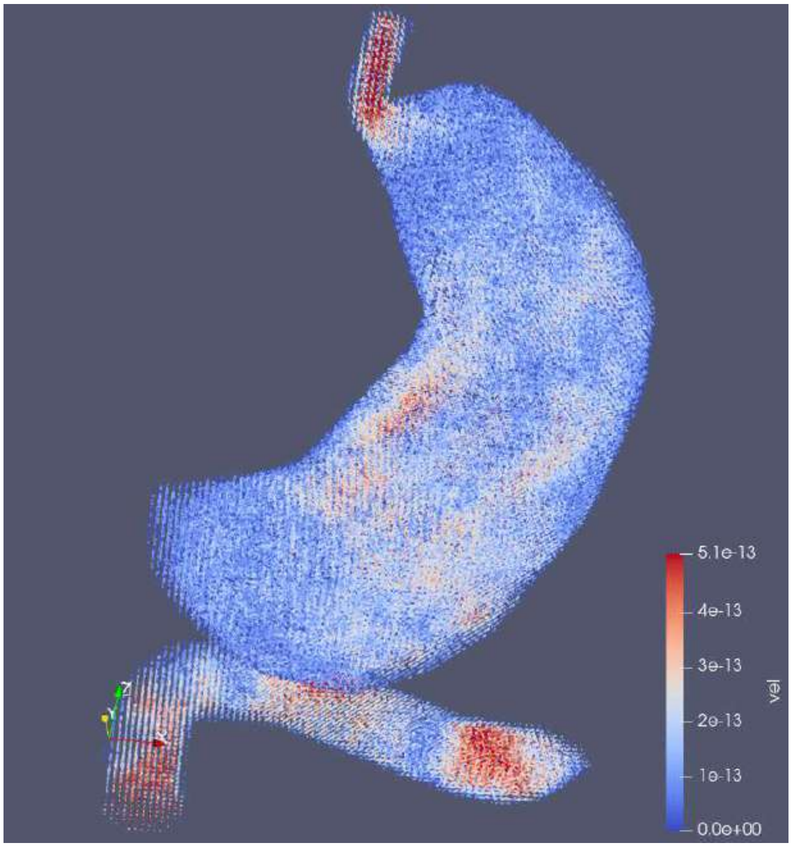


Fig 6. Velocity field distribution in anastomosis state at time 2,5 sec

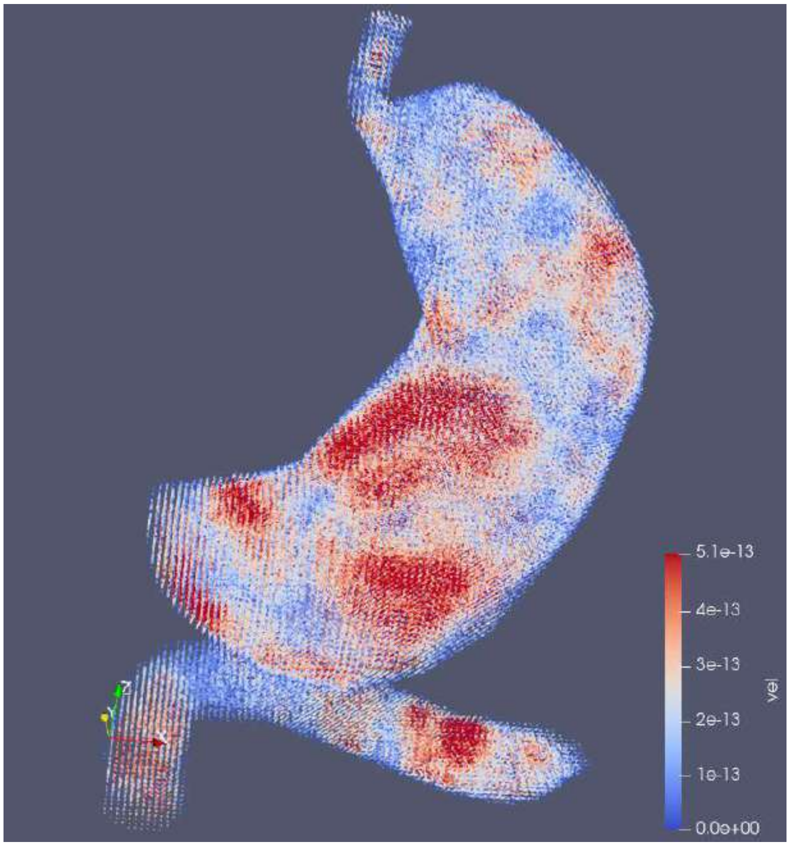


Fig 7. Velocity field distribution in anastomosis state at time 5,0 sec

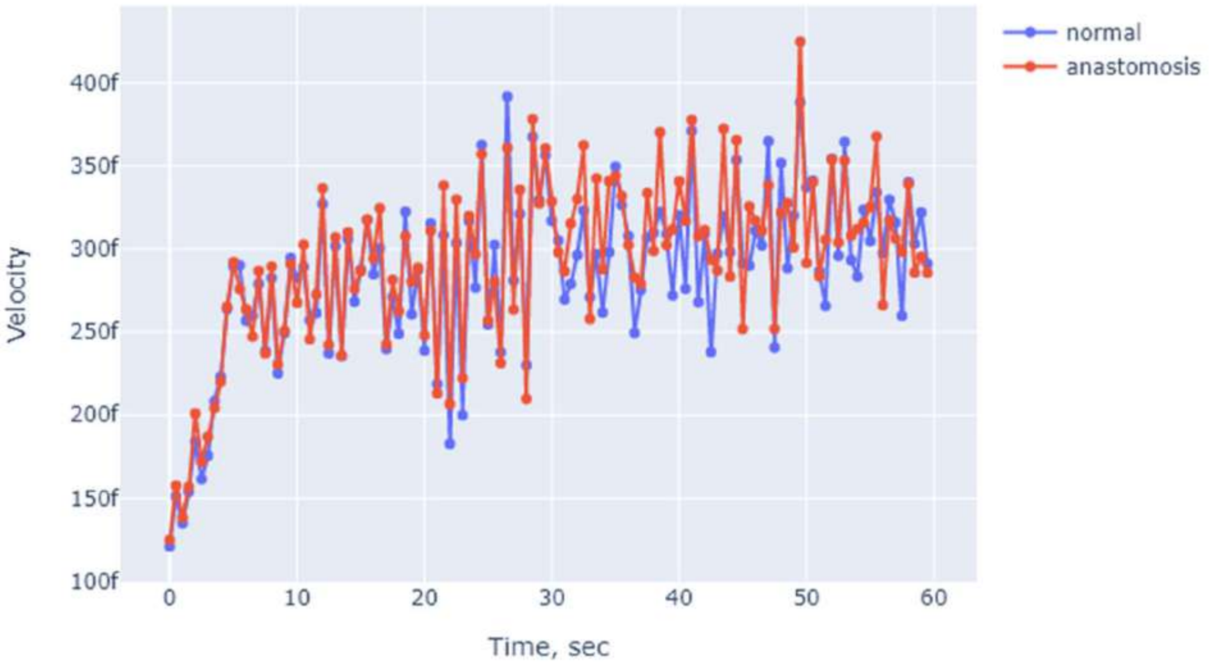


Fig.8 Average velocity during modeling

Conclusions

This paper studies the principles of simulating with the lattice Boltzmann models in fluid motion simulation on closed surfaces. The human digestive tract was chosen as an appropriate example of a closed surface due to the practical significance of this model.

This article studies the principles of using the lattice Boltzmann model for simulating the movement of fluid in objects with closed surfaces. Modeling is implemented on the example of the closed surface of the human digestive tract. Such studies are of great practical importance as they increase the results of reconstructive operations on the human digestive tract.

The developed simulation model provided a unique insight into the fluid dynamics of gastric contents. The conducted experiments show a clear difference in simulated behavior between the normal state of the stomach and the state of anastomosis. This result indicates the practical significance of our work. In addition, the proposed approach made it possible to analyze the processes in the digestive tract in dynamics by visualizing the pressure distribution and changes in the velocity field along the entire modeling geometry.

One of the possible implementations of the investigated method is detecting regions in the gastrointestinal tract where values of concerned fluid flow properties are higher or lower than some critical thresholds. It can help for better planning of surgery operations. The second possible application can be real-time monitoring of the gastrointestinal tract during the process or post-operation. All those implementations require accurate diagnostic tools, which can show the inner structure and geometry of the patient’s gastrointestinal tract. There is a possibility of transforming into a 3D model that can be handled by simulation software.

The presented approach to the dynamic simulation of fluid flows in closed surfaces of a complex shape has a particular drawback, which is associated with insufficient accuracy in determining changes in the pressure distribution. A further research direction is the application of machine learning technologies to increase this accuracy.

References

[1] S. Aoki, K. Uesugi, K. Tatsuishi, H. Ozawa, and M. Kayano, “Evaluation of the correlation between in vivo and in vitro release of phenylpropanolamine HCl from controlledrelease tablets,” *International Journal of Pharmaceutics*, vol. 85, no. 1, pp. 65–73, 1992, doi: https://doi.org/10.1016/03785173(92)90135O.

[2] K. Molly, M. Vande Woestyne, and W. Verstraete, “Development of a 5-step multi-chamber reactor as a simulation of the human intestinal microbial ecosystem,” *Appl Microbiol Biotechnol*, vol. 39, no. 2, pp. 254–258, May 1993, doi: https://doi.org/10.1007/bf00228615.

[3] M. J. Y. Yoo and X. D. Chen, “GIT Physicochemical Modeling – A Critical Review,” *International Journal of Food Engineering*, vol. 2, no. 4, Nov. 2006, doi: https://doi.org/10.2202/1556-3758.1144.

[4] S. Singh, “Fluid Flow and Disintegration of Food in Human Stomach,” 2007. doi: https://doi.org/10.13140/RG.2.2.11961.42084.

[5] A. Pal, K. Indireshkumar, W. Schwizer, Bertil Abrahamsson, M. Fried, and J. G. Brasseur, “Gastric flow and mixing studied using computer simulation,” *Proceedings of The Royal Society B: Biological Sciences*, vol. 271, no. 1557, pp. 2587–2594, Dec. 2004, doi: https://doi.org/10.1098/rspb.2004.2886.

[6] A. Pal, J. G. Brasseur, and Bertil Abrahamsson, “A stomach road or ‘Magenstrasse’ for gastric emptying,” *J Biomech*, vol. 40, no. 6, pp. 1202–1210, Jan. 2007, doi: https://doi.org/10.1016/j.jbiomech.2006.06.006.

[7] P. L. Bhatnagar, E. P. Gross, and M. Krook, “A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral OneComponent Systems,” *PR*, vol. 94, no. 3, pp. 511–525, May 1954, doi: https://doi.org/10.1103/PhysRev.94.511.

[8] X. He and L. Luo, “Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation,” *PRE*, vol. 56, no. 6, pp. 6811–6817, Dec. 1997, doi: https://doi.org/10.1103/PhysRevE.56.6811.

[9] “pylbm/pylbm,” *GitHub*. https://github.com/pylbm/pylbm

[10] T. Krüger, H. Kusumaatmaja, A. Kuzmin, O. Shardt, G. Silva, and E. M. Viggen, *The Lattice Boltzmann Method*. Cham: Springer International Publishing, 2017. doi: https://doi.org/10.1007/978-3-319-44649-3.

[11] Blender Foundation, “blender.org – Home of the Blender project – Free and Open 3D Creation Software,” *blender.org*, 2019. https://www.blender.org/

[12] M. Rast, “Simultaneous solution of the Navier-Stokes and elastic membrane equations by a finite element method,” *International Journal for Numerical Methods in Fluids*, vol. 19, no. 12, pp. 1115–1135, Dec. 1994, doi: https://doi.org/10.1002/fld.1650191205.

[13] T. Inamuro, M. Yoshino, and F. Ogino, “Accuracy of the lattice Boltzmann method for small Knudsen number with finite Reynolds number,” *Physics of Fluids*, vol. 9, no. 11, pp. 3535–3542, Nov. 1997, doi: https://doi.org/10.1063/1.869426.

[14] N. G. Deen, V. Sint, and J. A. M. Kuipers, “Detailed computational and experimental fluid dynamics of fluidized beds,” *Selected papers from the Third International Conference on CFD in the Minerals and Process Industries*, vol. 30, no. 11, pp. 1459–1471, 2006, doi: https://doi.org/10.1016/j.apm.2006.03.002.

[15] H. Başağaoğlu, J. R. Harwell, H. Nguyen, and S. Succi, “Enhanced computational performance of the lattice Boltzmann model for simulating micronand submicronsize particle flows and nonNewtonian fluid flows,” *Computer Physics Communications*, vol. 213, pp. 64–71, 2017, doi: https://doi.org/10.1016/j.cpc.2016.12.008.

[16] V. W. Azizi, G. Závodszky, B. J. M. van Rooij, and A. G. Hoekstra, “Inflow and outflow boundary conditions for 2D suspension simulations with the immersed boundary lattice Boltzmann method,” *Computers & Fluids*, vol. 172, pp. 312–317, Aug. 2018, doi: https://doi.org/10.1016/j.compfluid.2018.04.025.UDC 004.052.42

UDC 004.052.42 DOI: https://doi.org/10.20535/2708-4930.3.2022.269112

ZERO-KNOWLEDGE IDENTIFICATION OF REMOTE USERS BY UTILIZATION OF PSEUDORANDOM SEQUENCES

I. Daiko, V. Selivanov, M. Chernyshevych, O. Markovskyi

The article theoretically substantiates, proposes and investigates an identification scheme based on the concept of "zero knowledge" using irreversible generators of pseudorandom bit sequences. Session passwords form a chain generated by selective sequence values. Secondary identification sessions are provided in the proposed scheme to counter attacks with the displacement of one of the remote interaction parties. The main elements of the proposed identification scheme are developed in detail: authorization procedures, primary and secondary identification.

***Key words**: Zero-knowledge identification, chain of passwords, cryptographically strong identification, generators of pseudo-random bit sequences, middle attacks.*

Introduction

The high rate of progress in the technical means of the Internet, as well as the COVID-19 pandemic, have resulted in the rapid spread of remote interaction technologies. On the other hand, progress in the field of nanotechnology has stimulated the dynamic expansion of the use of remote computer control systems for real-world objects. A characteristic feature of such systems, which have received the name Internet of Things (IoT), is that the Internet is used as a data transmission medium [1]. The above-mentioned expansion of the use of remote information interaction systems to new areas of human activity stimulates the corresponding growth of computer crimes, the purpose of which is to affect the processes of data exchange between the parties of such interaction [2].

This requires adequate improvement and development of means of protection of processes of remote information interaction. Mechanisms for mutual identification of remote interaction participants occupy an important place among these tools. In recent years, there has been a significant qualitative and quantitative increase in attacks on these mechanisms, and new forms of violation of identification processes emerged. Accordingly, there is an objective need for the improvement of means of identification of the parties of remote interaction to ensure the proper level of data protection, differentiation of access rights to them, and the efficiency of implementing economic forms of the organization of remote provision of information services.

The problem of reliable identification is particularly acute for systems of computer remote control of objects in the real world using the Internet as a data transmission medium. The terminal devices of such remote-control systems are portable microcontrollers with built-in radio modems. These computing devices have low computing power, but must implement real-time identification procedures. New methods of fast and reliable identification need to be developed for them.

Thus, the scientific task of improving the effectiveness of means of identification of participants in remote information interaction is relevant and practically crucial given the peculiarities of the current stage of information technology development.

Problem statement and review of methods for its solution

The effectiveness of identification mechanisms, like any other means of cryptographic data protection, is characterized by two criteria [3]:

– the level of security, which is estimated by the number of resources needed to breach protection;

– the amount of resources for the implementation of protection functions. As the last criterion, the time of execution of protection functions on the computing platform of the participant of information interaction is most often used.

Traditionally, the task of identifying a remote participant in information interaction is one of the three fundamental tasks of modern cryptography [4]. The analysis of this problem is based on the classic model of remote information interaction. This model takes the presence of a system that remotely provides definite information services to a certain number of subscribers. This means that the specified classical model assumes an asymmetric nature of threats: the motivation to obtain illegal access to system resources is much higher than the motivation to provide services to the user instead of the system. Accordingly, within the framework of the classic model, the identification mechanisms are also asymmetric in terms of both the level of security and the speed of implementation of protective functions: for a system that serves thousands of subscribers, the identification time should be orders of magnitude longer. Such a model sufficiently correctly reflects a wide range of real multi-user systems, as well as a significant nature of threats to the information security of computer management systems of real-world objects [5]. An essential element of the model is that data exchange is carried out over potentially vulnerable Internet data transmission channels.

Within the classical model discussed above, the goal of attacks on identification mechanisms is to gain illegal access to system resources, to obtain information exchanged between the system and users, or to change it intentionally. The objects of the attack are the data exchange channel between the user and the system and, in particular, Internet switching centers [6].

A passive attack on the channel involves control of the data transmitted over the channel during the identification process. Active action on the channel involves interception of the data exchange session after the system identifies the user (middle attacks).

Another object of attack is the system in which the identification data of its users is stored. Technologically, the attack on the system is mainly carried out under the guise of a legal user, virus programs, or mafia fraud, that is, by influencing the system personnel [7].

All known methods of identification of remote participants of information interaction are divided into two classes: cryptographically strict and cryptographically weak [3].

Cryptographically strict identification methods must satisfy the following conditions:

– The password must be changed in each session of information interaction in order to protect against passive attacks on the channel and attacks on the system in which passwords are stored.

– The system should not store any information that allows for the reproduction of subscribers' passwords.

Accordingly, weak identification methods use permanent passwords that can potentially be intercepted during transmission in the channel and used to illegally penetrate the system under the guise of a legitimate user. A class of hybrid identification methods can be singled out separately, which fulfill only one of the above conditions. This class includes, in particular, the mechanism for identifying users of UNIX systems [3], within which only the second condition is satisfied.

In practice, cryptographically strict identification is implemented most often in the form of the concept of "zero knowledge" [8], which is based on two provisions:

– the subscriber must have a cryptographic mechanism for generating correct passwords;

– the system must have at its disposal a cryptographic mechanism for checking the correctness of passwords, which, however, does not allow the system itself to generate correct passwords.

Until now, a wide range of means of identification has been proposed within the framework of the concept of "zero knowledge" [9 – 12], which uses various cryptographic mechanisms for the formation of correct passwords and their verification by the system.

Existing methods of cryptographically strict identification can be divided into two classes:

– with unrelated session passwords;

– with session passwords related to each other.

In the identification schemes of the first type, mathematical multi-valued irreversible transformations are used as a mechanism for checking the correctness of the subscriber's password. In the well-known Guillou-Quisquater [9], Schnorr[10], and FESIS [11] schemes, irreversible number theory transformations are used as such transformations. In these schemes, the impossibility for the system to independently generate the correct session passwords is due to the analytical intractability of the discrete logarithm problem. The possibility of using a large number of independent session passwords is because this problem has an infinite set of solutions [3].

On the other hand, the use of number theory problems to build a mechanism for verifying the correctness of a password has the consequence of spending considerable time on their implementation, given the high computational complexity of performing modular exponentiation of large-bit numbers.

A particular increase in identification speed while preserving the above-mentioned cryptographic properties can be achieved using the algebra of finite Galois fields [12], especially in hardware implementation.

Significantly greater opportunities for fast cryptographically strict identification are provided by schemes with associated session keys. In fact, if when using independent session passwords, the user proves that he is the same one who registered in the system, then with dependent passwords, he demonstrates that he is the same one who interacted with the system in the previous information interaction session.

A classic identification scheme of this type [13] involves the use of an irreversible hash transformation. At the registration stage, the subscriber forms a chain of session passwords, each resulting from a hash transformation over the previous one. Accordingly, the subscriber uses these passwords in reverse order, so the system has the last password session as an identification code. Due to the irreversibility of the hash conversion, it cannot determine the next session password. Using such a scheme provides 3 – 4 orders of magnitude faster identification than the methods discussed above based on irreversible transformations of number theory.

An analysis of the current practice of attacks on remote interaction systems shows that when using cryptographically strict identification schemes, the biggest threat is middle attacks [14]. These attacks are carried out after the subscriber's identification is completed and consist in pushing him away from informational interaction with the system.

The most effective way to counter these types of attacks is to carry out repeated identification cycles during the information interaction session. The review showed that the most significant disadvantage of known cryptographically strict identification schemes in current conditions is vulnerability to middle attacks.

Purpose and objectives of research

The purpose of the work is to increase the effectiveness of cryptographically strict identification of participants in remote information interaction due to the acceleration of the identity confirmation process, as well as by organizing secondary cycles of contact control to counteract interaction interception.

To achieve the set goal, the following tasks are solved in the work:

– analysis of the possibilities of use for cryptographically strict identification of the fastest–acting standardized cryptographic mechanisms – generators of pseudo-random binary sequences;

– development and research of a method of cryptographically strict identification, which is distinguished by the use as a mechanism for checking the correctness of the session password on the side of the system of properties of irreversible generators of pseudorandom binary sequences, due to which, an increase in speed is achieved and the possibility of implementing a series of secondary accelerated identification cycles using them;

– theoretical and experimental evaluation of the effectiveness of using generators of pseudo-random binary sequences as a mechanism for checking the correctness of the session password in terms of speeding up the identification process, as well as increasing the level of security.

The object of research is the process of cryptographically strict identification of participants in remote information interaction, which provides the possibility of protection against session interception by outsiders.

The method of implementing the concept of “zero knowledge” using pseudo-random sequences for subscriber identification

In current conditions and in the future, the level of security of identification processes acceptable for most applied applications can be achieved only by applying the progressive cryptographic concept of "zero knowledge." However, the main problem with the practical use of these technologies is the need for significant dusting resources to implement the corresponding cryptographic transformations.

A high speed of cryptographically strict identification can be achieved only when nonlinear Boolean transformations are used as irreversible transformations. It is well known that the system of nonlinear Boolean equations cannot be solved by analytical methods [3]. The only way to solve such systems of nonlinear Boolean equations is to perform a complete enumeration.

Cryptographic transformations implementing the concept of "zero knowledge" using non-linear Boolean functions can be carried out by three known mechanisms:

– one–way hash transformations;

– cipher blocks that are used in the mode of one-way transformations;

– generators of binary pseudorandom sequences used in stream ciphers.

It is well known that current ciphers [4] provide the highest speed of cryptographic protection of information. They are widely used for real-time encryption and decryption of telephone conversations and transmission of video images over closed channels. The main advantage of using pseudo-random sequences as irreversible transformations for implementing the cryptographic concept of "zero knowledge" is a significantly faster performance than hash transformations and cipher blocks.

The developed method of rapid identification of remote interaction participants involves using a generator of pseudorandom bit sequences by the system and each remote subscriber. Cryptography uses generators that remember the state, non-linear Boolean functional transformations of the transition to the next state, and the formation of the output bit. In other words, the generator circuit fits into the well-known abstract automaton model [15]. The fundamental point here is that the Boolean functional transformations used have high nonlinearity and meet the criterion of the avalanche effect. This makes it impossible to reconstruct the sequence of bits of the line using the methods of linear and differential cryptanalysis [8]. Unlike the traditional one, the automaton model of the generator of pseudorandom bit sequences has no input signals; that is, the mathematical model of the generator is an abstract Moore automaton. This means that the line of following the generator states is determined uniquely with fixed settings of the feedback functions. At the same time, the feedback functions are organized so that the line of states has a maximum period of τ and includes all possible conditions. This means the generator of pseudorandom binary sequence forming a bit sequence of $B = b_1, b_2, \dots b_\tau, b_1, b_2, \dots$ with a repetition period τ.

The use of pseudorandom binary sequences in modern cryptographic data protection mechanisms is based on the practical difficulty of restoring the entire series by its k-bit fragment b_1, b_2, …, b_k. The complexity of this problem is due to the nonlinearity of the output bit formation function based on the status code of the generator, which transforms the task of restoring the sequence into a system of nonlinear Boolean equations that cannot be solved analytically. This means that the only way to restore the series by its fragment is an enumeration, which with current values of the period τ goes far beyond the possibilities of technical implementation. In a practical sense, this means that knowing the algorithm of the pseudorandom bit sequence generator for its given fragment, it is impossible to restore the state of the generator, starting from which the given segment is generated.

The developed method of cryptographically strict identification of participants of remote information interaction involves the procedures of system subscriber registration, identification at the beginning of the session, and secondary identification, which is carried out periodically during the current session.

The first of the mentioned procedures is that a chain of m session passwords is formed, stored in the subscriber's memory, and its first element is sent to the system that performs identification. The procedure is performed in the following order:

1. The subscriber randomly generates the code S_m of the generator status of pseudo-random sequences on the m-th identification session. The setting of the Boolean function of the output signal's formation is performed, as well as the setting of the feedback functions of the generator.

2. The selected code S_m is loaded into the generator's memory, after which a sequence of k bits is formed, which form the session password S_{m-1}.

3. The value of m decreases by one: $m = m - 1$. If after that $m > 0$, a return is made to the execution of the previous item 2.

4. The pseudo-random bit sequence generator setting codes chosen by the subscriber and the S_0 code are encrypted with the system's public key and sent to it. The generated sequence of codes S_1, S_2, …, S_m of the generator state is stored in the subscriber's memory.

5. The system receives the registration message from the subscriber, decrypts it with its private key, and stores the value of the codes of the subscriber's generator settings and the S_0 code in the memory area allocated for the subscriber's service.

The developed subscriber identification procedure at the beginning of the i-th session of information interaction, $i \in \{1, 2, \ldots, m\}$ includes the following actions:

1. The subscriber sends the i-th S_i code to the system, which plays the role of a session password. In addition, the subscriber initiates the operation of the generator of pseudo-random bit sequences with the start code S_i and forms a line with a length of $k + d$ bits. The first k bits of the generated series constitute the X code, and the last d bits set the U code.

2. The remote system receives an initialization code from the subscriber via the Internet, by which it selects the setting codes for the generator to work with the subscriber from memory. Then the system adjusts the generator by the read codes. After receiving the session password code Si from the subscriber, the system initiates the operation of the generator of pseudo-random bit sequences with the start code S_i and forms a line of length $k + d$ bits. The first k bits of the formed series form the Y code, and the last d bits form the W code.

3. The system compares the Y code generated from the generator operation with the S_{i-1} code of the previous session password stored in the memory. Suppose these codes are identical, i.e., $S_{i-1} = Y$. In that case, the primary identification of the subscriber in the current session is considered successful, and system resources determined by his status are provided to him. The S_{i-1} code in the system memory is replaced by the accepted session password S_i. In addition, the system sends the user the code W generated by it. If $S_{i-1} \neq Y$, the system sends a zero code to the subscriber.

4. The subscriber receives the access rights granting code from the system: if it is zero, the system has not identified it. Otherwise, the received code W is compared by the subscriber with the code U: if $U = W$, then the user gets confirmation that he has a remote interaction session with the system. In this case, he starts an information interaction session.

The developed procedure for the basic identification of the subscriber by the system corresponds to the above criteria of the theoretical concept of "zero" knowledge. The first of these criteria is satisfied because the session password codes S_1, S_2, …, S_m used by the subscriber are all different. In practice, this is guaranteed by the appropriate choice of their length – k. It is also quite evident that the system, having the S_{i-1} code at its disposal, is not able to obtain the code of the next session password, even knowing the settings of the Boolean functions of the output signal formation and the feedback function of the generator of pseudo-random bit sequences. It is not difficult to show that the specified problem in the mathematical sense is identical to the solution of a system of k nonlinear Boolean equations, which cannot be solved analytically. Thus, the second criterion of the concept of "zero" knowledge is fulfilled: the system cannot generate the subscriber's correct session password.

To protect against attempts to intercept the process of information interaction between the system and the subscriber after its identification by blocking at the switching centers of global networks, the procedure for intermediate identification of the participants of the information interaction has also been developed within the framework of the proposed method, which makes it possible to periodically check the identity of the participants of the information interaction. This procedure consists of the following sequence of actions:

1. The system periodically, without additional generator setting, forms a $2{\cdot}d$-bit pseudo-random sequence, the first d bits of which form the Q code and the last d-the G code. The system sends the G code obtained in this way via the Internet to the subscriber with whom the information interaction is carried out.

2. The subscriber receives the G code from the system and initiates re-identification. For this, the subscriber, without additional adjustment of his generator, generates a $2{\cdot}d$-bit pseudo-random sequence, the first d bits of which form the V code, and the last d bits constitute the C code. After that, the subscriber compares the generated code C, and the code G received from the system: if $G = C$, then this means that the system supports information interaction with it. In this case, the subscriber sends the V code he generated to the system.

3. The system receives the V code from the subscriber and compares it with the Q code generated using the system generator. If these codes match, i.e., $V = Q$, then the system makes sure that the information interaction takes place in the subscriber and that he was not pushed out of the session by an intruder.

Thus, the use of the described procedure for secondary identification of participants in remote information interaction allows you to reliably detect the presence of attacks on a session after its start. It is quite obvious that the resource costs for secondary identification are an order of magnitude lower compared to primary identification. This allows for secondary identification during the session without appreciable impact on the speed of data transfer between participants.

Effectiveness evaluation of the method

The main criteria for the effectiveness of systems for identifying participants in remote information interaction are the level of security and the speed of technical implementation of protection procedures.

The task of breaking the proposed method of cryptographically strict identification is identical to the task of predicting a pseudorandom binary sequence. When using standardized cryptographic generators of pseudo-random bit sequences, this task is one whose practical implementation is beyond technical capabilities [4]. In order to break the security of the system, that is, to simulate access to it by a legitimate user, it needs to determine two components: the number h_x of sequence generation steps and the initial state Q_x. This can only be done by sorting.

The possibility of implementing basic and repeated identification cycles using a single mechanism – a cryptographic generator of pseudorandom bit sequences – is the main advantage of the proposed method of cryptographically strict identification. Reputed cryptographically strict identification schemes [9 – 12] do not provide such a possibility. Another significant practical benefit of the developed technique of cryptographically rigorous identification compared to noted ones is much faster performance.

In the software implementation of the SHA-256 standardized hash transformation, the most common in practice, 64 cycles are performed, in each of which 6 shifts, 15 arithmetic addition operations, 5 logical AND operations, 6 logical XOR operations are performed, i.e. a total of 2048 operations.

With the software implementation of the AES cipher block in the minimum configuration, 10 cycles are performed, in each of which the logical XOR operation of the data block with the key is performed (4 processor operations), 16 operations of accessing the byte substitution tables of the data matrix, 3 cyclic shift operations for shuffling the rows of this matrix, 8 shift and logical addition operations to shuffle the columns of the data matrix. The total number of processor operations is therefore 310 processor operations.

Structures of generators of pseudorandom generators are much simpler and dozens of processor operations are used to generate one bit, which is much less compared to cipher blocks or hash transformations.

It is proved [3] that block ciphers and hash converters, which are used in known cryptographically strict identification schemes, have a lower speed order of magnitude than generators of pseudorandom bit sequences.

Unlike known methods, only one pass of a password is used for a single identification session over potentially dangerous data lines.

The proposed method of cryptographically strict identification allows to detect the fact of an attempt to oust a legal participant of information interaction from the session. The attacker does not know the settings of the pseudorandom sequence generator, so he cannot generate the correct sequence of bits that confirms the presence of an information contact. However, the developed method does not protect the process of remote interaction for types of attacks in which the attacker fully controls the information flows between the interaction parties at the switching center. To implement such protection, it is necessary to use a generator of pseudo-random sequences for stream encryption of data exchanged by participants of information interaction. The use of a generator of pseudorandom binary sequences within the framework of the proposed solution allows the use of a single cryptographic mechanism for data identification and encryption.

Conclusion

Conducted research aimed at increasing the effectiveness of cryptographically strict identification of participants in remote information interaction allowed to obtain the following results:

It has been established that the main shortcomings of existing schemes of cryptographically strict identification in modern conditions are insufficient speed, as well as the inability to resist new types of attacks, in particular, displacement of the subscriber from the process of remote information interaction after his identification by the system.

A method of cryptographically strict identification has been developed and researched, which is distinguished by the use as a mechanism for checking the correctness of the session password on the system side of the properties of irreversible generators of pseudorandom binary sequences, due to which we achieve an increase in performance and the possibility of implementing a series of secondary accelerated identification cycles using them;

The proposed method of accelerated identification of participants of remote information interaction is oriented for use in real-time computer control systems of remote objects using global networks.

References

[1] M. M. Noot and W. H. Hassan, "Current research on Internet of Things (IoT) ," *Compute Network*, vol. 148, no. 15, pp. 283–294, 2019.

[2] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, no. 5, pp. 395–411, May 2018, doi: https://doi.org/10.1016/j.future.2017.11.022.

[3] B. Schneier and W. Diffie, *Applied cryptography: protocols, algorithms, and source code in C*. Indianapolis (Ind.): Wiley, Cop, 2015, p. 784.

[4] A. J. Menezes, P. C, and S. A. Vanstone, *Handbook of applied cryptography*. Boca Raton: CrcPress, 1997, p. 780.

[5] I. Mashal, O. Alsaryrah, and T. Y. Chung, "Analysis of recommendation algorithms for Internet of Things," in *2016 IEEE Wireless Communications and Networking Conference*, pp. 1–6. doi: https://doi.org/10.1109/WCNC.2016.7564667.

[6] K. Kittichokechai and G. Caire, "Secret KeyBased Identification and Authentication with a Privacy Constraint," *IEEE Transactions on Information Theory*, vol. 62, no. 11, pp. 6189–6203.

[7] M. Han, Z. Yin, P. Cheng, X. Zhang, and S. Ma, "Zero-knowledge identity authentication for internet of vehicles: Improvement and application," *PLOS ONE*, vol. 15, no. 9, pp. 217–247, Sep. 2020, doi: https://doi.org/10.1371/journal.pone.0239043.

[8] N. Bardis, N. Doukas, and O. P. Markovskyi, "Fast subscriber identification based on the zero knowledge principle for multimedia content distribution," *International Journal of Multimedia Intelligence and Security*, vol. 1, no. 4, pp. 363–377, 2010.

[9] J. J. Quisquater and Louis C., "A Practical Zero-Knowledge Protocol Fitted to Security Microprocessor Minimizing Both Transmission and Memory," *Eurocrypt-88*, vol. 330, pp. 123–128, 1988.

[10] C. P. Schnorr, "Method for identifying subscribers and for generating and verifying electronic signatures in a data exchange system," 1989 Available: https://patents.google.com/patent/US4995082A/en

[11] U. Feige, A. Fiat, and A. Shamir, "Zero-knowledge proofs of identity," *Journal of Cryptology*, vol. 1, no. 2, pp. 77–94, Jun. 1988, doi: https://doi.org/10.1007/bf02351717.

[12] N. G. Bardis and N. Doukas, "A Method for strict remote user authentication using non-reversible Galois field transformations," in *MATEC Web of Conferences*, 2017, pp. 243–249.

[13] Y. ASIMI, A. AMGHAR, A. ASIMI, and Y. SADQI, "Strong Zero-Knowledge Authentication Based on the Session Keys (SASK)," *International Journal of Network Security & Its Applications*, vol. 7, no. 1, pp. 51–66, Jan. 2015, doi: https://doi.org/10.5121/ijnsa.2015.7105.

[14] M. Conti, N. Dragoni, and V. Lesyk, "A Survey of Man in The Middle Attacks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2027–2051, 2016, doi: https://doi.org/10.1109/comst.2016.2548426.

[15] T. Unkašević, Z. Banjac, and M. Milosavljević, "A Generic Model of the Pseudo-Random Generator Based on Permutations Suitable for Security Solutions in Computationally-Constrained Environments," *Sensors*, vol. 19, no. 23, p. 5322, Dec. 2019, doi: https://doi.org/10.3390/s19235322.UDC 004.052.42

UDC 004.052.42 *DOI: https://doi.org/10.20535/2708-4930.3.2022.269132*

ORGANIZATION OF PROTECTED FILTERING OF IMAGES IN CLOUDS

A. Mirataei, O. Rusanova, K.Tribynska, O. Markovskyi

The article proposes an approach to using cloud technologies to accelerate the filtering of image streams while ensuring their protection during processing on remote computer systems. Homomorphic encryption of images during their remote filtering is proposed to be carried out by shuffling rows of pixel matrices. This provides a high level of protection against attempts to illegally restore images on computer systems that filter them. The developed approach makes it possible to speed up the performance of this important image processing operation by 1 – 2 orders of magnitude.

Key words*: Arithmetic mean filtration, images processing, homomorphic encryption, secure clouds computing.*

Introduction

One of the defining features of the current stage of information technology development is the rapid process of qualitative improvement of the interface between computer systems and the outside world. Image processing, analysis and recognition is a key element of computer perception of objects in the outside world. Solving these problems includes a number of stages, one of which is improving the quality of the image and updating it, that is, removing elements from it that do not carry useful information for solving a specific problem of image analysis. The main tool of the image updating process is its filtering. In addition to performing the task of updating, image filtering ensures an increase in their quality by removing the interference that occurred during image reception and transmission [1].

In practice, two types of filtering are used: median and arithmetic mean. Both types of filtering involve scanning the image with an aperture of a certain size and are characterized by significant resource consumption, proportional to the product of the number of pixels in the image by the square of the aperture size. On the other hand, image filtering procedures allow simultaneous processing of several of their fragments. This determines the feasibility of using multiprocessor computer systems for image filtering [2].

For most practical applications of image analysis, it must be performed in real-time on terminal low-power microcontrollers. This dictates the need to involve remote powerful computer systems using the capabilities of modern cloud technologies to perform resource-intensive analysis and image recognition operations [3]. Currently, the vast majority of terminal microcontrollers are equipped with built-in radio modems, which makes it possible to connect them to the Internet. A significant obstacle to the use of cloud technologies for the radical acceleration of image processing is that, for a significant part of practical applications, the condition of confidentiality must be fulfilled, which excludes the possibility of access to the images by third parties who can potentially use this information to disrupt the operation of computer health monitoring systems objects of the real world. Based on this, the task of protecting images in the process of their processing and, in particular, filtering, on remote computer systems arises. In practical terms, we are talking about homomorphic encryption of images. which makes it impossible to illegally reconstruct them on remote computer systems performing filtering.

Thus, the scientific task of ensuring the protection of images in the process of their filtering on uncontrolled remote multiprocessor computing systems is relevant and practically important for the current stage of development of computer technologies.

Problem statement and review of methods for its solution

The need for significant computing resources is characteristic of all image processing, analysis, and recognition tasks. This is due to the fact that modern images consist of millions of dots, and the continuous process of improving their quality results in an increase in the number of dots. In addition, modern image processing algorithms are constantly becoming more complex. Thus, trends in the use of computer image analysis in modern conditions dictate the need for a significant increase in the amount of computing resources, the growth rates of which significantly outpace the progress of increasing the speed of processors [1]. With this in mind, the most effective way to speed up computer image processing is to use cloud technologies that provide access to practically unlimited computing resources [4].

The main obstacle to the implementation of this possibility is that for the vast majority of practical applications it is unacceptable to transfer images to potentially accessible remote computer systems. Therefore, for the possibility of remote processing and analysis of images, it is necessary to carry out their homomorphic encryption, which allows processing with the possibility of decrypting the obtained results. For estimate of effectiveness of such type encryption is used followed criterias [5].

Acceleration of the implementation of filtering due to the use of cloud systems, which is estimated by the coefficient γ. The value of this coefficient is determined by the ratio of the time T_0 of filtering the image on the terminal device to the time T_{cd} of its performing homomorphic encryption of the image and decryption of the obtained results:

$$\gamma = \frac{T_0}{T_{ed}} \quad .(1)$$

The level of security of images when using homomorphic encryption, which is estimated by the amount of resources needed to carry out illegal image restoration on a remote computer system.

In the last decade, results were obtained [6,7], which show the fundamental possibility of creating homomorphic ciphers invariant to processing procedures. However, the computational complexity of the samples of such ciphers created so far makes their practical use impractical.

Therefore, in practice, specialized homomorphic ciphers adapted to certain data processing procedures are used. For the tasks of secure filtering of images on remote computer systems, a number of specialized homomorphic ciphers have also been proposed to date [8,9]. The vast majority of them are based on additive masking of image pixels. The essence of additive masking is that when masking an image, which is given by the matrix $B=\|b_{i,j}\|$, $i\in\{1,2,\ldots n\}, j\in\{1,2,\ldots,m\}$, an image is created-mask, which is given by the matrix $V==\|v_{i,j}\|$. When performing median filtering with an aperture of odd size h, its central element $b_{i+(h+1)/2,j+(h+1)/2}$ is replaced by the median h^2 of the aperture pixels. It is obvious that the additive masking should be performed in such a way that the order between the values of the aperture pixels is not violated, that is, the code of each pixel of the mask $v_{i,j}$ should depend on the corresponding code of the original image. This significantly reduces the effectiveness of additive masking due to the fact that significant computing resources are spent on homomorphic encryption and decryption. With arithmetic mean filtering, the central element $b_{i+(h+1)/2,j+(h+1)/2}$ of the aperture is replaced by the arithmetic mean of its pixels:

$$\forall i = 1,2,..,n, j = 1,2,..,m : b_{i,j} = \frac{1}{h^2}\cdot\sum_{l=1}^{h}\sum_{q=1}^{h} b_{i+l,j+q} \ . \quad (2)$$

The operation of additive encryption consists in adding to each pixel of the original image the corresponding pixel of the mask: $u_{i,j} = b_{i,j} + v_{i,j}$. On the remote computer system, filtering of the masked image U is performed, during which a new value $u_{i,j}$ is formed in the form:

$$u_{i,j} = \frac{1}{h^2}\cdot\sum_{l=1}^{h}\sum_{q=1}^{h} u_{i+l,j+q} = \frac{1}{h^2}\cdot\sum_{l=1}^{h}\sum_{q=1}^{h} b_{i+l,j+q} + \frac{1}{h^2}\cdot\sum_{l=1}^{h}\sum_{q=1}^{h} v_{i+l,j+q} \ . \quad (3)$$

It follows from formula (3) that the result of remote filtering is the sum of the result of filtering the original image B and the mask V. Accordingly, homomorphic encryption consists in subtracting from each pixel of the resulting image U the code of the pixel of the same name as the result of mask filtering.

The main disadvantage of homomorphic ciphers based on additive masking is that it requires an additional mask image filtering procedure. In paper [10], it is proposed to randomly select one of the previously filtered images as a mask. For a wide range of practical applications, the stream of processed images is sufficiently correlated, which significantly reduces the level of security of images during their remote processing. Therefore, a significant drawback of the additive masking method using a permanent mask or one of the previously filtered images is that it does not provide a high level of security. Accordingly, the disadvantage of the known method of homomorphic encryption of images during their remote filtering is an insufficiently high level of security.

Purpose and objectives of research

The purpose of the work is to increase the efficiency of secure image processing in the clouds, in particular, their arithmetic mean filtering on remote computer systems by increasing the level of security.

The main tasks of the research in accordance with the set goal are as follows.

1. Analysis of computational operations of arithmetic mean filtering, identification of homomorphic ciphers invariant to the filtering procedure, and selection of the most effective of them for further development.

2. Development of a method of homomorphic encryption of images based on string permutations to increase the level of security during remote arithmetic mean filtering.

3. Theoretical and experimental evaluation of homomorphic encryption efficiency indicators based on changing the order of lines during arithmetic mean filtering on remote computer systems.

The object of research is the processes of homomorphic encryption of images for their protected arithmetic mean filtering in clouds.

Method of homomorphic encryption of image upon arithmetic mean filtration

The conducted analysis of the possibilities of increasing the efficiency of protected filtering of images in the clouds allows us to conclude that the most promising way to achieve the goal is to use mixing. This procedure, traditional for cryptography, does not require significant computing resources and can be performed directly during image transmission. In favor of such a conclusion, the fact that the significant amount of information contained in modern images makes shuffling a sufficiently effective means of protection against attempts to reconstruct images using technologies of directed enumeration or statistical analysis. Based on this, the basis of the proposed method of homomorphic encryption of images during their arithmetic mean filtering is their mixing.

Within the framework of the developed method, the original image *B* is transformed into an encrypted image *U* by rearranging its lines in a certain order. At the same time, the order of permutation of the rows of the image matrix *B* is used as a key for its homomorphic encryption. After homomorphic encryption, the image *U* is sent to a remote computer system where it is partially filtered. The image *R* obtained as a result of such filtering is returned to the terminal microcontroller, which performs homomorphic decoding of *R* by restoring the order of lines, and also performs the final filtering phase.

The order of shuffling the rows of matrix *B* is chosen arbitrarily and fixed in the form of a table *Q* of direct permutation. From the table *Q*, the table *G* of the reverse permutation is formed, so that the condition *a=G(Q(a))* is fulfilled. Table *G* is used for homomorphic decryption of remote processing results.

The proposed method of homomorphic encryption of images for their protected arithmetic mean filtering on remote computer systems involves the periodic operation of generating tables of forward *Q* and reverse *G* permutation of matrix rows of image pixels.

The process of remote secure processing of the image specified by the matrix *B*, according to the proposed method, consists of the following sequence of actions:

1. According to table *Q*, the rows of the matrix *B* are permuted, in the result matrix *U* of the encrypted image is formed:

$$U = \begin{pmatrix} u_{11}\ u_{12} \ldots u_{1m} \\ u_{21}\ u_{22} \ldots u_{2m} \\ \ldots. \\ u_{n1}\ u_{n2} \ldots u_{nm} \end{pmatrix}.$$

2. The *U* matrix of the shuffled image obtained as a result of the homomorphic encryption described above is sent to a remote computer system.

3 On a remote computer system, a partial arithmetic average filtering of the elements of the matrix *U* is performed. The result of this operation is formed in the form of a matrix *C*:

$$C = \begin{pmatrix} c_{11}\ c_{12} \ldots c_{1m} \\ c_{21}\ c_{22} \ldots c_{2m} \\ \ldots. \\ c_{n1}\ c_{n2} \ldots c_{nm} \end{pmatrix}.$$

The procedure of partial arithmetic mean filtering provided by the developed method consists in replacing each element $u_{i,j}$ of the matrix *U* divided by h^2 by the sum of *h* elements of the fragment of the *i*-th row of the matrix, such that its central element belongs to the *j*-th column of the matrix:

$$\forall i \in \{(h+1)/2,..,n-(h-1)/2\},\ j \in \{(h+1)/2,...,m-(h-1)/2\} : c_{i,j} = \frac{1}{h^2} \cdot \sum_{l=j-h/2}^{j+h/2} u_{i,l}\ . \qquad (4)$$

4. Formed as a result of partial arithmetic mean filtering on remote computer systems, matrix *C* is returned to the computer platform that performs image processing and analysis.

5. Over the received matrix *C*, the reverse permutation of rows is performed using table *G*; as a result, the matrix *F* is formed.

6. On the matrix *F*, the operation of the final stage of filtering is performed, which consists in the fact that each element $f_{i,j}$ of the matrix *F* is replaced by the sum of h elements of the fragment formed by the *j*-th column, and the central element of this fragment is the element belonging to the *i*-th row matrices *F* :

$$\forall i \in \{(h+1)/2,..,n-(h-1)/2\},\ j \in \{(h+1)/2,...,m-(h-1)/2\} : f_{i,j} = \sum_{l=i-h/2}^{i+h/2} f_{l,j}\ . \qquad (5)$$

The image *F* formed by the described procedure is a filtered original image *B* with an aperture equal to *h*.

Evaluation of the developed method effectiveness

Evaluation of the effectiveness of the developed and proposed method of protected remote arithmetic mean filtering can be carried out according to the criteria specified in the review section.

Acceleration of the implementation of filtering due to the use of cloud systems is estimated by the coefficient γ. To determine it, it is necessary to determine the spent time T_0 filtering the image on the terminal device, as well as the time T_{cd} for it to perform homomorphic encryption of the image and decrypt the results received from the cloud. When performing filtering on the terminal microcontroller of the image processing system, h^2 addition operations and one division operation must be performed for each of the $n \cdot m$ image points. Considering that the execution time of the division command is about 50 times longer than the addition operation [11], the numerical value of T_0 can be estimated by the expression: $T_0 \approx n \cdot m \cdot (h^2 + 50) \cdot t_a$, where t_a is the execution time of the addition command on the terminal microcontroller.

The time T_{cd} for the execution of encryption, decryption and the final filtering stage by the terminal microcontroller is determined by calculating the following. The proposed method for homomorphic encryption of images before their transmission to the cloud does not involve special calculations, as it is reduced to permuting the rows of the pixel matrix. Technologically, the process of permuting the rows of the matrix is reduced to changing the order of transferring image points to the network and, accordingly, does not require additional time. That is, the process of homomorphic image encryption in reality can be combined with the process of data transmission from the terminal microcontroller to a remote computer system.

Similarly, the process of homomorphic decoding of a partially filtered image, which is carried out by reverse shuffling of the image pixel matrix rows using table *G*, can be combined with the process of transferring a partially filtered image from a remote computer system to a terminal microcontroller. This means that in the proposed method of their protected arithmetic average filtering of images in the cloud, the decryption process also does not require additional time resources of the terminal microcontroller. This property of permutation ciphers combined with the remote nature of image processing, which requires cycles of transmission over the Internet, is the main factor in the increased efficiency of homomorphic encryption compared to other known methods of specialized homomorphic encryption in terms of image processing acceleration.

Thus, the time T_{cd} when using the proposed method is determined by the time required to perform the final stage of filtering, which, in accordance with formula (5), consists in adding h numbers from the matrix F for each pixel of the specified matrix. Realistically, when scanning the matrix of images, it is more expedient not to recalculate the sum, but to perform only two operations: to subtract the value of the element that goes beyond the boundaries of the fragment during scanning and to add the value of the element that entered the boundaries of the fragment. That is, the process of processing each point of the image in the process of the final stage of its filtering requires only two arithmetic operations. In other words, the numerical value of time T_{ed} is determined by the product $T_{ed} = n{\cdot}m{\cdot}2{\cdot}t_a$. Accordingly, the value of the coefficient γ of accelerating the implementation of filtering due to the use of cloud systems is calculated as:

$$\gamma = \frac{T_0}{T_{ed}} = \frac{m \cdot n \cdot t_a \cdot (h^2 + 50)}{m \cdot n \cdot 2 \cdot t_a} = \frac{1}{2} \cdot (h^2 + 50)\,. \tag{6}$$

For example, with a value of $h = 15$ typical for practical applications, calculated according to formula (6), the value of the coefficient γ of accelerating the implementation of filtering due to the use of cloud technologies is 137,5. According to experimental research, the value of the coefficient γ is slightly smaller and is about 130. The obtained value of the coefficient γ practically coincides with the estimation of acceleration of filtering for the fastest known variants of homomorphic encryption of images by the method of additive masking.

When applying the proposed method of secure image filtering on remote computer systems, the time required to process each point of the image consists of the time of performing h operations of arithmetic addition, as well as the time of performing the operation of division by the square of the aperture size h^2 of the received sum using one division command . In other words, the specific weight υ of operations performed on the terminal microcontroller during the implementation of the final stage in the total volume of arithmetic mean filtering operations is determined by the following expression:

$$\upsilon = \frac{T_{ed}}{T_T} = \frac{n \cdot m \cdot 2 \cdot h \cdot t_a}{n \cdot m \cdot t_a \cdot (h + 50)} = \frac{2}{h + 50}, \tag{7}$$

where T_T is the processing time on the terminal microcontroller of one point during partial arithmetic mean filtering of the image. At the value h=15 typical for real applications, the value of the specific weight υ of operations performed on the terminal microcontroller is υ = 0.0307 or 3.07%, calculated by formula (7). For additive masking of images with their protected processing in the cloud, this indicator is 4.5%.

The analysis of formula (7) indicates that in the proposed method of homomorphic encryption of images, only about 3% of the volume of calculations related to filtering is carried out on the terminal microcontroller, and, accordingly, 97% of the volume of calculations is implemented on remote computer systems. Thus, the conducted analysis proved that compared to other known methods of homomorphic encryption of images with their remote arithmetic mean filtering, the proposed method based on permutations of rows of the image matrix has better time indicators compared to the known method of additive masking. This effect is achieved due to the fact that the proposed method is based on permutation operations, which can be combined in practice with data transfer processes from the terminal microcontroller to cloud systems and back.

However, the main advantage of the proposed method of protecting images from their illegal reconstruction during filtering on remote computer systems is a significantly higher level of security.

Known methods of homomorphic encryption of images for their protected arithmetic mean filtering, in particular, methods based on additive masking, actually have to use the same mask for image processing, for which arithmetic mean filtering needs to be performed on the user's computing platform. The use of one mask for homomorphic encryption of several images makes it possible to detect this by means of spectral analysis. Accordingly, the party carrying out the attack has the opportunity to restore the masking image and, accordingly, decipher the real image in the process of its processing on a remote computer system not controlled by the user.

The level of security of an image can be estimated by the amount of resources required by the party aiming to restore the original image. When using the developed method of homomorphic encryption of images for their protected arithmetic mean filtering, the number *d* of possible permutations of rows of the pixel matrix is *n*!. For real images, the number of n rows of the pixel matrix is 1024, respectively, the number of d options for permuting the columns of such an image is d=1024! = $6.421\cdot10^{2639}$. It is clear that the analysis of such a large number of options makes it practically impossible to restore the original image by selecting the reverse permutation, since the selection of such a significant number of options is far beyond the scope of technical implementation with modern computer means. The implementation of such a large number of options is impossible even in the near term of 20 – 40 years, even if the capabilities of quantum computers are used.

For certain classes of contour images, the volume of the considered search can be significantly reduced due to directional image reconstruction. This technology involves selecting rows in such a way that two adjacent ones are minimally different from each other. The conducted experimental studies showed that for real contour images, the volume of the search can be reduced by 2 – 3 orders of magnitude in this way. But it is quite clear that reducing the sorting order by 0.02% does not significantly affect the technical implementation of such sorting by modern computer means.Even with the application of the described technology, the selection of such a significant number of options is far beyond the scope of modern technical implementation capabilities. When processing images of these classes using the developed method, it is recommended to organize simultaneous filtering of a group of *k* images. At the same time, *n* rows of *k* images can be rearranged according to the proposed technology within the group. The number of images within one group does not affect the time taken to encrypt the image before sending it to the network and the final filtering phase. However, the number of permutation options increases to $(n\cdot k)!$, which very effectively increases the level of security of images. for example, even with k=2, the number of row sorting options is $1.67\cdot10^{5894}$.

Conclusion

As a result of research aimed at increasing the level of security of images when filtering them on remote computer systems, a new method of homomorphic encryption was theoretically substantiated, developed and researched.

The proposed method of homomorphic encryption of images to protect against their illegal reconstruction during arithmetic mean filtering on remote computer systems differs in that the main element of protection is the shuffling of image pixel matrix rows. The shuffling order can change randomly and serves as a secret key for homomorphic encryption of images. Within the framework of the developed method, procedures for partial arithmetic mean filtering, which is carried out on remote systems, as well as procedures for the final stage of filtering are defined, which is carried out on a terminal platform that performs processing and analysis of a real image. The developed method of protected filtering based on shuffling the rows of the pixel matrix allows, due to the use of remote computing power, to speed up this operation by 1 – 2 orders of magnitude, which practically coincides with the similar indicator of the fastest-acting variant of image protection based on additive masking.

The main advantage of the developed method is a much higher level of protection against attempts, using statistical analysis, to gain illegal access to images during their processing on remote computer systems not controlled by the user.

The proposed method can be used to speed up the processing and analysis of images by terminal devices of computer systems for remote monitoring of the state of real-world objects and their management.

References

[1] J. C. Russ and Neal Brent F, *The image processing handbook*. Boca Raton, Fla.: Crc, 2017, p. 1053.

[2] O. P. Markovskyi, A. M. Bilashevska, and M. O. Nevdashenko, “Protected implementation of image filtration on GRID systems,” *Visnik of National Technical University of Ukraine “KPI” Informatics, Control and Computer Engineering*, no. 61, pp. 105–109, 2014.

[3] V. A. Sathish and T. A. Sangeetha, “Cloud-based Image Processing with Data Priority Distribution Mechanism,” *Journal of Computer Applications*, vol. 6, no. 1, pp. 6–8, 2013.

[4] M. M. Boroujerdi and S. Nazem, “Cloud Computing: Changing cogitation about computing,” *IJCSI International Journal of Computer Science Issues.* vol. 9, no. 3, pp. 169–180, 2012.

[5] O. P. Markovskyi, I. O. Humeniuk, Alireza Mirataei, Ya. I. Toroshanko, and M. O. Voloshchuk, “METHOD FOR SPEED UP PROTECTED IMAGE FILTRATION IN CLOUDS,” *Telekomunìkacìjnì ta ìnformacìjnì tehnologìï*, vol. 4, no. 65, pp. 99–110, Jan. 2019, doi: https://doi.org/10.31673/2412-4338.2019.049911.

[6] van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, “Fully Homomorphic Encryption over the Integers,” in *Advances in Cryptology – EUROCRYPT 2010*, H. Gilbert, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 24–43.

[7] C. Gentry and S. Halevi, “Implementing Gentry’s FullyHomomorphic Encryption Scheme,” in *Advances in Cryptology – EUROCRYPT 2011*, K. G. Paterson, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 129–148.

[8] O. P. Markovskyi, I. O. Humeniuk, Alireza Mirataei, Ya. I. Toroshanko, and M. O. Voloshchuk, “METHOD FOR SPEED UP PROTECTED IMAGE FILTRATION IN CLOUDS,” *Telekomunìkacìjnì ta ìnformacìjnì tehnologìï*, vol. 4, no. 65, pp. 99–110, Jan. 2019, doi: https://doi.org/10.31673/2412-4338.2019.049911.

[9] M. Ahmed and Mohammad Ashraf Hossain, “Cloud Computing and Security Issues in the Cloud,” *International Journal of Network Security & Its Applications*, vol. 6, no. 1, pp. 25–36, 2014, doi: https://doi.org/10.5121/ijnsa.2014.6103.

[10] I. O. Humenuk, “Method removed Arithmetic mean filtration of images,” *Almanac of sciense,* vol. 11, no. 32, pp. 40–43, 2019.

[11] A. Subero, *Programming PIC Microcontrollers with XC8*, 1st ed. CA: Apress Berkeley, 2018, p. 434. doi: https://doi.org/10.1007/978-1-4842-3273-6.

UDC 004.056.5 DOI: https://doi.org/10.20535/2708-4930.3.2022.265479

FAST SECURE CALCULATION OF THE OPEN KEY CRYPTOGRAPHY PROCEDURES FOR IOT IN CLOUDS

A. Mirataei, M. Haidukevych, O. Markovskyi

The article proposes a method for accelerating the implementation of cryptographic data protection mechanisms on embedded IoT terminal microcontrollers, the basic operation of which is the modular exponentiation of high-capacity numbers. The method is based on the use of remote computer systems to speed up calculations and provides protection against the reconstruction of the secret keys of cryptosystems from data transmitted to the cloud. It has been theoretically and experimentally proven that the method allows, on average, 50 times, to speed up the implementation of cryptographic data protection protocols in IoT while providing a level of security sufficient for most practical applications.

***Key words**: Modular exponentiation, secure cloud computing, IoT security, RSA cryptosystems.*

Introduction

The dynamic development of Internet technologies led to the emergence and rapid spread of remote-control systems for real-world objects. Such systems are called the Internet of Things. Real-world objects are directly controlled using portable embedded microcontrollers that are equipped with radio modems for Internet communication with a central computer. According to [1], today the number of such microcontrollers significantly exceeds the number of personal computers.

At the same time, there is a steady trend towards expanding the scope of the Internet of Things. For most areas of application of the Internet of Things, the issue of ensuring reliable protection against external influence is critical. The potential threat of such influence is due to the fact that data exchange is carried out directly through the open Internet. Accordingly, to ensure reliable protection of the Internet of Things, it is necessary to realize the possibility of fully using cryptographic protocols for information protection [2].

The vast majority of existing information protection protocols involve the use of public key cryptography. The basic computational operation of such cryptography is modular exponentiation, which is performed with large numbers. In particular, currently this operation uses 2048-bit numbers with the prospect of growth to 4096 in the coming years. Performing a modular exponentiation operation on numbers of such a digit size requires a significant amount of computing and time resources. Embedded portable low-bit microcontrollers of the Internet of Things do not have enough computing power to implement such complex calculations in real time. In recent years, the most acceptable way to solve the problem of the shortage of computing power is to use cloud technologies [3]. Such technologies provide access to significant computing resources of remote computer systems, including those that have modular exponentiation hardware.

Direct use of these resources by microcontrollers of the Internet of Things to implement cryptographic protocols is impossible, since these calculations contain secret keys. Accordingly, there is a need for such an organization of modular exponentiation, in which the majority of the volume of calculations is performed on a remote computer system, while it is practically impossible to recover the secret keys based on the data provided to it.

Thus, the scientific task of developing a method of secure modular exponentiation in the cloud to accelerate the implementation of cryptographic protection protocols in the Internet of Things is relevant for the current stage of information technology development.

Problem statement and review of methods for its solution

In recent years, the use of cloud technologies has become the dominant approach to solving the problem of the shortage of computing power in solving applied problems [3]. These technologies allow using the Internet to provide a wide range of users with significant computing power of modern multiprocessor computer systems to quickly solve their applied problems. In fact, within the framework of cloud technologies, the effect of virtualization of the availability of significant computing power to users is achieved. This approach not only increases the capabilities of users by orders of magnitude, but also ensures the economic efficiency of creating high-capacity computer systems [4].

One of the main disadvantages of cloud technologies, which significantly limits their practical application, is the possibility of unauthorized access to user data during their processing on remote computer systems [5].

Therefore, in recent years, work on the creation of homomorphic data encryption methods has been carried out on a broad front [6,7]. Unlike traditional data encryption, which is widely used in data transmission and storage, homomorphic encryption allows you to perform certain types of processing of encrypted data with the possibility of restoring the correct processing results through decryption [8]. Until now, schemes of the so-called full homomorphic encryption have been proposed, in which multiplication and addition operations can be performed on the encrypted data [9]. An important point here is that full homomorphic data encryption actually has the properties of universality, as it can be applied to a sufficiently wide range of data processing tasks, which are reduced to addition and multiplication operations. However, the existing fully homomorphic encryption schemes have significant computational complexity and are not yet in practical use, although quite intensive research is being conducted to create fully homomorphic encryption methods acceptable for practical use [10].

In practice, specialized homomorphic encryption schemes are used to protect data and the process of their processing on remote computer systems, which are focused on a specific procedure for remote data processing. Accordingly, for the operation of modular exponentiation $A^{E} mod M$ - the basic operation of modern cryptography, specialized methods of protection of secret elements, *A* and *E,* have been developed [11].

The vast majority of known methods of protected modular exponent calculation are based on the additive decomposition of the exponent code *E* into components, part of which is used for exponentiation on remote computing power, and part of which is used to perform this operation on the terminal microcontroller. This allows you to protect the code of the exponent *E* from reconstruction attempts based on the data that is transmitted to the cloud. Other mechanisms are used to protect against attempts to reproduce the number *A* [12].

In [12], a method of remote calculation of the modular exponent based on the random division of the exponent code *E* into groups of digits is proposed. This allows the computation of $A^{E} mod M$ to be organized as a modular product of modular exponents that can be computed independently on multiple remote computer systems. To protect the number *A*, which is raised to a power, its multiplication by a secret number is used, which is selected in such a way that the result of its modular raising to the power *E* is equal to one.

A significant advantage of the considered method is that the capabilities of multiprocessor systems can be fully used for the parallel calculation of partial modular exponents. In [13], based on theoretical and experimental data, it is shown that the described method can actually increase the performance of modular exponentiation by approximately three times.

The work [14] describes the method of secure calculation of the modular exponent based on the logical decomposition of the code of the exponent *A*. This approach allows you to speed up the calculation of partial exponents by reducing the number of modular multiplication operations by a constant number. At the same time, it is possible to transfer intermediate data from remote computer systems, which allow to significantly speed up the calculation of the components of modular exponentiation on the terminal processor. To protect the number *A*, its multiplication by a special number is applied, for which a modular inversion is determined in advance.

It is known about a group of methods for the secure calculation of the modular exponent, which is based on the fact that the multiplicative components of the module *M* are known, which makes it possible to organize a secure calculation by adding to the code of the exponent a number that is a multiple of the Euler secret period [15].

Another interesting method is described in [16]. The main emphasis in these studies is on the fact that the user can not only remotely perform modular exponentiation in closed mode, but also indirectly control the correctness of the performed operations.

The analysis of modern technologies for secure computation of the modular exponent on remote computer systems showed that their significant drawback is that different cryptographic mechanisms are used to protect both secret components of this operation.

Purpose and objectives of research

The aim of the study is to increase the speed of implementation of cryptographic information protection mechanisms on terminal microcontrollers of computer control systems in real time by organizing the secure execution of the basic operation of these mechanisms – modular exponentiation on remote computer systems.

To achieve the set goal, the following tasks are solved in the framework of this work:

– a review of existing methods for the secure calculation of the modular exponent on remote computing systems, an analysis of the possibilities for improving their efficiency;

– development of a method for accelerating the calculation of the basic operation of a wide range of cryptographic algorithms with a public key – modular exponentiation on terminal microcontrollers through the use of remote powerful computer systems based on the multiplicative-additive decomposition of the exponent code, which is the secret key of cryptosystems;

– theoretical and experimental evaluation of the effectiveness of the developed method based on the criterion of the level of protection from the reconstruction of secret components to data transmitted to the cloud, as well as the criterion of the achieved acceleration of the implementation of cryptographic data protection protocols on low-power portable terminal microcontrollers of computer control systems for real world objects.

The method of protected modular exponentiation in the cloud based on multiplicative-additive exponential decomposition

To achieve the goal, the method of protected modular exponentiation A^E mod M on remote computing capacities has been developed, which is based on the multiplicative-additive decomposition of secret code of the exponent E, that is its representation in the form $E=F\cdot a+k$. At the same time, the F code is transmitted to the remote system, and the components k and a are used only on the terminal microcontroller. The main advantage of the proposed approach is the possibility of simultaneously solving two problems: the practical impossibility of illegal reconstruction of the secret codes of the exponent E and the number A by the code F and the open value of the module M.

The expansion of the exponent is carried out by selecting two random parameters a and k. Moreover, the number a is chosen so that its bit rate m_a does not exceed 5, and the value k must satisfy the condition $(E-k)$ mod a =0. Since the exponent code E is part of the private key, in real public-key information security protocols, the numbers E and M rarely change, so they can be considered constant. This allows you to perform the selection of parameters a and k described above only once during key generation.

The proposed method of secure remote calculation of the A^E mod M modular exponent involves the following sequence of actions:

1. The modular exponentiation operation $G=A^a$ mod M is performed on the terminal microcontroller.

2. The result of the calculation of G, as well as the numbers F and M are sent to a remote computer system.

3. The calculation of the modular exponent $W= G^F$ mod M is implemented in the cloud and the result is returned via the Internet to the terminal microcontroller.

4. At the same time, the modular exponent $Y=A^k$ mod M is calculated on the terminal microcontroller.

5. After obtaining the W value from the cloud, the modular product of the W and Y values is calculated on the terminal microcontroller: $W\cdot Y$ mod M.

The operation of the proposed method of protected modular exponentiation in the cloud based on the multiplicative-additive expansion of the exponent can be illustrated by the following numerical example. Let the module M=143, as well as the public key D=7 and the private key E=103 be formed at the stage of generating a cryptosystem with a public key. The latter is used as a secret key of the terminal microcontroller of the system of remote control of objects in the real world. The next step is to expand the exponent: $E=F\cdot a+k$. Let the chosen value of a be equal to 3. According to the above, the value of variable k should be selected in such a way that $(E-k)$ mod a=0. One of the possible options satisfying the condition can be $k = 4$, since (103 – 4) mod 3 =0.

As part of the current example, the modular exponent 80^{103} mod 143=115 is calculated, respectively A=80, E=103, M=143. According to clause 1, the auxiliary quantity $G=A^a$ mod $M = 80^3$ mod 143=60 is calculated. In accordance with clause 2, the obtained number G=60 and the values F=33 and M=143 are sent to a remote computer system to calculate the modular exponent $W=G^F$ mod $M=60^{33}$ mod 143=125. The result of W=125 is returned to the terminal microcontroller. At the same time, using the terminal microcontroller, the operation of modular elevation of the number A to the k power is implemented: $Y=A^k$ mod $M=80^4$ mod 143=81. According to clause 5, the modular product of the values of W and Y is calculated: $R=W\cdot Y$ mod M =125·81mod143 =115. The final result of the calculations is the code R=115.

As follows from the above, the proposed method allows you to quickly calculate the basic operation of public key cryptography on a low-power portable terminal microcontroller through the use of powerful remote computer systems. At the same time, no codes are transmitted to these systems that allow recovering the values of the secret codes of the modular exponentiation operation. An important element of the proposed method is that the operation of modular exponentiation is also implemented on remote computer systems. This makes it possible to use cryptoprocessors for the fast implementation of this operation, which allow speeding up the execution of modular exponentiation by 2 – 3 orders of magnitude.

The constructiveness of the proposed method can be proved as follows. Considering that the value calculated in the cloud is $W=G^a$ mod $M = ((A^F)^a$ mod $M = A^{F\cdot a}$ mod M. Operations performed on the terminal microcontroller can be represented as: $Y=A^k$ mod M, $R=W\cdot Y$ mod $M = A^{F\cdot a}\cdot A^k$ mod $M = A^{F\cdot a+k}$ mod M. Since $E=F\cdot a+k$, that $R=A^E$ mod M, which was to be proved.

The developed method differs in that a single mechanism is used to encrypt the secret code of the exponent and the number A raised to a power the multiplicative-additive decomposition of the exponent code. The use of a single cryptographic mechanism for encrypting the two secret components of the modular exponentiation operation makes it possible to increase the proportion of operations performed on remote computer systems. As a result, a greater acceleration of this operation, which is important for cryptographic applications, is achieved in comparison with known approaches. At the same time, a high level of security is maintained, based on the analytically unsolvable discrete logarithm problem.

The above feature determines the originality of the completed development. Another important feature of the proposed method is that it allows you to simultaneously perform the remote exposure operation on encrypted data, which is carried out on remote computer systems, and the exposure operation using an additive component on the terminal microcontroller. This solution makes it possible, on the one hand, to increase the speed of implementation of cryptographic protection mechanisms with a public key, and, on the other hand, allows choosing the value of the additive component of the exponent code within a wide range. The latter provides a higher level of protection provided against brute force attempts to select the value of the additive component. In known methods, expanding the range of possible values of the additive component of the exponential code necessarily entails a decrease in the computational speed. In the proposed scheme, the expansion of the range of possible values of the additive component due to the organization of parallel computing has practically no effect on the time of computing the modular exponent.

Evaluation of the developed method effectiveness

It is advisable to evaluate the effectiveness of the proposed method according to two criteria: determining the performance indicators and the level of protection against the attempts of the villain to reconstruct the value of the exponent E and the number A based on the data transmitted to the cloud.

The level of protection is determined by the amount of time resources that the villain must spend on trying to recover the secret code of the exponent and the number A. Since the public key D is known, the attacker can determine the values of X and U such that $X^E \bmod M=U$ by calculating $U^D \bmod M = X$. Accordingly, to find the secret code of the exponent, the most appropriate tactic of the attacker is the selection of unknown parameters a and k. It is quite obvious that the T_{CR} time for implementing such a selection depends on the number of possible options for the values of parameters a and k, and the time T_{EXPcl} – the calculation of the modular exponent in the cloud. The numerical value of T_{CR} is determined by the formula:

$$T_{CR} = 2^{m_a+m_k} \cdot T_{EXP\_CL} \tag{1}$$

It follows from formula (1) that the required level of protection, that is, the amount of time resources for its violation, can be ensured by the appropriate selection of the value of the parameter k.

The technology of such flexible management of the security level when using the proposed method can be illustrated by the following example. Suppose that gaining access to the private key E of the terminal microcontroller of the remote object control system allows the villain to obtain V_{SRC} benefits valued at \$1,000,000. Accordingly, to ensure reliable protection, it is necessary that the C_{CS} cost of the amount of resources spent on selecting parameter k exceeds the benefits, that is, it should be at least \$1,000,000. Assuming that the cost of renting the C_{CS} time of the remote computer system is \$100/hour, the specified condition is fulfilled for T_{CR} values determined from the equation:

$$T_{CR} = \frac{V_{SRC}}{C_{CS}} = 10^4 \ hours \ . \tag{2}$$

Determining the numerical value of the $\mathrm{T_{EXP\_CL}}$ time for calculating the modular exponent can be done based on the fact that nowadays almost all computer systems, including laptops, have hardware to quickly perform this important operation in the form of a built-in cryptoprocessor. In particular, the Hi/fn 7955 cryptoprocessor provides modular exponentiation of 2048-bit numbers in the time T_{EXP_CL}=0.083 s.= $2.3{\cdot}10^{-5}$ hours. Taking into account the determined numerical values of T_{CR} and T_{EXP_CL} included in formula (1), it is possible to obtain:

$$m_k + m_a = \log_2 \frac{T_{CR}}{T_{EXP\_CL}} = \log_2 \frac{10^4}{2.3 \cdot 10^{-5}} = 29\,. \tag{3}$$

Hence, the number m_k of binary digits of the number k is determined as 29 – 2 =27. That is, to ensure the level of protection defined in the current example, the bit number k must be at least 27.

As an indicator of speed, the acceleration factor β is most often used, which is determined by the ratio of the time T_M of calculating the modular exponent on the terminal microcontroller to the time T_C of performing this operation with the involvement of remote computing power according to the proposed method:

$$\beta = \frac{T_M}{T_C}\,. \tag{4}$$

When calculating the modular exponent by the classic method, the calculation time is T_M=$1.5{\cdot}n{\cdot}T'$, where n is the number of bits of the exponent, and T' is the time of modular multiplication of n-bit numbers [7]. Under the condition of using the Montgomery scheme for modular multiplication, the time T' is $2{\cdot}n^2{\cdot}t/r$, where r is the bit rate of the processor, and t is the time the processor executes one instruction of the addition type. Thus, the formula for calculating the modular exponent has the following form:

$$T_M = \frac{3 \cdot n^3 \cdot t}{r}. \tag{5}$$

It is clear from this formula (4) that there is a cubic dependence of T_M time on bit rate n.

The time T_C of calculating the modular exponent according to the proposed method depends on three components: the duration of the operation $A^a \mathrm{mod} M$ on the terminal microcontroller, the maximum value between the time of calculating the modular exponent in the cloud and the time of implementing the modular raising of the number A to the k power, as well as the time T' of the modular multiplication $R=W \cdot Y \mathrm{mod} M$. Accordingly, the value of T_C can be represented in the form of a formula:

$$T_C = T_{EXP\_a} + \max(T_{CL}, T_{EXP\_k}) + T'. \tag{6}$$

The duration of T_{EXPa} according to the classical method of calculating the modular exponent is T_{EXPa}=1.5·m_a·T'. Similarly, the execution time of the A^k mod M operation is T_{EXPk}=1.5·m_k·T'.

When implementing the modular exponentiation operation on a remote computer system, the T_{CL} calculation time is:

$$T_{CL} = 2 \cdot T_{DT} + T_{EXP\_CL}, \tag{7}$$

where 2·T_{DT} is the time of transferring data to the cloud and returning results, T_{EXP_CL} is the time of G^F mod M execution. In fact, the duration of calculations in the T_{expF} cloud can be neglected due to the fast execution of calculations, therefore T_{CL}=2·T_{DT}.

Let max(T_{CL}, T_{EXPk})=T_{EXPk}, then the T_C calculation formula can be written as follows:

$$T_C = 1.5 \cdot (m_k + m_a) \cdot T'. \tag{8}$$

Then the acceleration coefficient β can be represented as:

$$\beta = \frac{1.5 \cdot n \cdot T'}{1.5 \cdot (m_k + m_a) \cdot T'} = \frac{n}{m_a + m_k}. \tag{9}$$

In the framework of the above example, the secret code E of the exponent has a bit size of 2048: n=2048. As described above, the bits m_k and m_a are 24 and 5, respectively. Therefore, β = 2048 / 29 = 70.6. That is, within the framework of the considered example, the application of the developed method provides acceleration of modular exponentiation by 70.6 times.

The conducted experimental studies showed that the real speedup achieved by using the proposed method ranges from 50 to 100, depending on the requirements for the level of security.

Conclusion

As a result of the research aimed at improving the efficiency of information protection in computer control systems operating in real time and using the Internet as a data exchange medium, the following results were obtained.

A method for accelerating the calculation of the basic operation of a wide range of cryptographic algorithms with a public key-modular exponentiation on terminal microcontrollers through the use of remote powerful computer systems is proposed and investigated. The method provides for the organization of protection of data transmitted to the cloud due to the multiplicative-additive decomposition of the exponent code, which is the secret key of cryptosystems.

It has been theoretically and experimentally proved that the developed method makes it possible to reliably protect the secret components of modular exponentiation when this operation is remotely implemented on potentially open computer systems. At the same time, the developed method makes it possible to speed up the implementation of cryptographic data protection protocols on low-power portable terminal microcontrollers of computer control systems for real world objects by almost two orders of magnitude.

The developed methods are focused on use in real-time computer control systems and can significantly speed up the implementation of cryptographic data protection protocols on low-power terminal microcontrollers of such systems.

References

[1] M. M. Noot and W. H. Hassan, "Current research on Internet of Things (IoT) ," *Compute Network*, vol. 148, no. 15, pp. 283–294, 2019.

[2] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, no. 5, pp. 395–411, May 2018, doi: https://doi.org/10.1016/j.future.2017.11.022.

[3] G. Fox and D. Gannon, *Using Clouds for Technical Computing*. Amsterdam: IOS Press, 2018, pp. 81–102. Available: https://ebooks.iospress.nl/volume/cloud-computing-and-big-data

[4] Mehrdad Mahdavi Boroujerdi and Soheil Nazem, "Cloud Computing: Changing Cogitation about Computing," *Zenodo (CERN European Organization for Nuclear Research)*, vol. 9, no. 4, pp. 169–180, Oct. 2009, doi: https://doi.org/10.5281/zenodo.1071009.

[5] L. Zhang, Y. Cui, and Y. Mu, "Improving Security and Privacy Attribute Based Data Sharing in Cloud Computing," *IEEE Systems Journal*, vol. 14, no. 1, pp. 387–397, doi: https://doi.org/10.1109/JSYST.2019.2911391.

[6] C. Kaur, H. M. Mourad, and S. S. Banu, "Security and Challenges using Clouds Computing in Healthcare Management System," *International Journal of Trend in Scientific Research and Development*, vol. 6, no. 3, pp. 44–52, 2019.

[7] V. Getov, "Security as a Service in Smart CloudsOpportunities and Concerns," in *2012 IEEE 36th Annual Computer Software and Applications Conference*, pp. 373–379. doi: https://doi.org/10.1109/COMPSAC.2012.112.

[8] V. Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully Homomorphic Encryption over the Integers," i*In Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, 2010, pp. 24–43.

[9] Z. Brakerski, "Fully homomorphic encryption without modulus switching from classical GapSVP," in *Annual Cryptology Conference*, Springer, 2012, pp. 868–886.

[10] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," *In International Conference on the Theory and Application of Cryptology and Information Security*, Springer, 2017, pp. 409–437.

[11] Z. Liu, H. Kim, I. Verbauwhede, J. Großschadl, H. Seo, and S. S. Roy, "Efficient ring-LWE encryption on 8-bit AVR processors," *In International Workshop on Cryptographic Hardware and Embedded Systems*, Springer, 2015, pp. 663–682.

[12] N. Bardis and O. Markovskyi, "Secure Implementation of Modular Exponentiation on Cloud Computing Resources," in *Proceeding of International Conference Applied Mathematics, Computational Science and Systems Engineering*, Athens, Greece, 2017, pp. 90–96.

[13] J. V. Kostenko and O. V. Rusanova, "Method Protected Modular Exponentiation on Remote Computers Systems," *Bulletin of the National Technical University of Ukraine "KPI". Informatics, management and computer techniques.*, no. 64, pp. 51–54, 2016.

[14] N. Bardis, "Green Implementation of Modular Arithmetic Operations for IoT and Cloud Applications," in *Green IT Engineering: Components, Networks and System Implementation*, 2017, pp. 43–64.

[15] O. Markovskyi, N. Bardis, N. Doukas, and S. Kirilenko, "Secure Modular Exponentiation in Cloud Systems," in *Proceedings of the Congress on Information Technology, Computational and Experimental Physics (CITCEP 2015)*, 2015, pp. 266–269.

[16] X. Chen, J. Li, J. Ma, Q. Tang, and W. Lou, "New Algorithms for Secure Outsourcing of Modular Exponentiations," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 9, pp. 2386–2396, Sep. 2014, doi: https://doi.org/10.1109/tpds.2013.180.

UDC: 004.02, 004.2 DOI: https://doi.org/10.20535/2708-4930.3.2022.265229

METHODS OF EFFECTIVIZATION OF SCALABLE SYSTEMS: REWIEW

O. Honcharenko, H. Loutskii

The article discusses the problem of inefficiency of modern systems and horizontal scaling as a method of increasing productivity. The main issues that make up the mentioned problem are highlighted. A classification for possible solutions was proposed, according to which they were divided into architectural and network, and an overview was carried out. As part of the architectural class, such approaches as quantum computing and the dataflow paradigm were reviewed, the most promising solutions were analyzed. The comparative analysis shows that by their nature dataflow and quantum computing do not contradict each other, moreover, they complement each other in the context of the problem. At the same time, both types of processors require a certain network for communication, which makes the issue of topology relevant. At the network level, 2 topologies – Fat Tree and Dragonfly – were considered, and their main properties were highlighted. The analysis showed that in the context of the problem Dragonfly is slightly better due to decentralization and smaller diameter. In the conclusions, the main aspects of problem formulation and review are indicated, further prospects and possible methods are considered.

***Keywords**: effectivization, scalable systems, high performant computing, architecture, topology*

Introduction

There are 2 main methods of scaling – horizontal and vertical. Vertical scaling is associated with increased performance of individual nodes, but this method has several disadvantages. First, it requires updating all nodes of the system, which is impractical from the point of view of finances. Secondly, the performance of individual processors is based on the clock frequency, the effective maximum of which is limited due to the complexity of the internal structure of the processor itself.

Another approach, horizontal scaling allows you to increase the peak speed linearly. However, there are problems here too. First, Amdahl's law says that dependencies between parallel parts of the task limit its parallelism. Thus, the dependence between the scope of the task, the scope of the system and the acceleration is established. Secondly, the properties of the system itself and the delays that occur in the execution process limit the real user acceleration. So, when the processor cannot continue execution, it must block the task, resulting in idle time. The solution to this problem is to switch to another task, but context switching requires significant time costs. As a result, a significant part of the calculations is occupied by idle time and interruptions. At the same time, energy consumption is proportional to the number of nodes, which raises the question of the feasibility of scaling in each specific case. The only way out of this situation is the modernization of the architecture and computing model: the search for such solutions that would allow either to gain in speed or to reduce costs. This makes the issue of efficiency *urgent*.

Reasons for limitation

The problem of efficiency is complex. This is not a single issue – it is a complex of issues that need consideration, analysis and resolution. Of course, their complete analysis is a separate topic for research, but even superficially analyzing the situation, one can highlight the key limitations of modern parallel supercomputer solutions.

1. *Problem of parallelism*. As mentioned earlier, this problem is that the classical model of computing has certain disadvantages that arise from the architecture of the processor.

2. *Problem of parallel programming*. The fact is that each parallel system is unique and has its own architectural features. At the same time, the task of parallelizing the algorithm is the prerogative of the programmer, which makes the development of each program difficult, and the program itself is architecturally tied to the system (or series of systems) under which it was written.

3. *The problem of inconsistency between the algorithm and the system*. This problem is related to specialized algorithms (for example, AI problems or genetic algorithms). These tasks have a high

degree of internal parallelism, but often during their execution there are restrictions dictated by the system architecture, which limits this internal parallelism. This raises the question of adapting not only the task to the system, but also the system to the task being performed.

4. *The problem of the balance of speed and costs*. A large number of computing facilities allows for acceleration, but the hardware comes at a price and consumes energy. A small amount of equipment limits consumption, but reduces performance. The question arises – how to achieve a balance of resources for each specific task within the framework of the system.

5. *The problem of node interaction*. A large number of nodes in the system leads to the fact that the only way to connect them is the internal network. But at the same time, a number of issues arise, such as the issue of fault tolerance, the issue of bandwidth, the issue of load balancing and the issue of the topological structure of the network as such. Both real performance and the possibility of practical application of the system as such depend on the effectiveness of their solution.

Analysis of the subject area

To find potential solutions, it makes sense to conduct a brief analysis of the subject area. This will allow you to highlight areas that need to be reviewed. Since the subject area is not monolithic, but consists of parts, the considered solutions cannot be called directly focused on it – they are directed only at specific aspects of it, but collectively they can give the key to a general solution. What aspects of the area can be highlighted and what exactly should be included in the review? In order to answer this question, it is necessary to clearly define which parts of the system the action of this or that method is aimed at. In general, the following classes of possible solutions can be distinguished:

1. *A change in architecture or computing paradigm*. Within the framework of this class, it is assumed that the Von Neumann architecture will be abandoned and a transition to other hardware and a different model of calculations will be made. At the same time, both hardware and software of the system are completely changed.

2. *Changing the principles of communication*. Within the framework of this class, the structure of individual computing elements (processors, nodes) is not considered – instead, the object of influence is the means of communication (communication lines, switches), communication protocols, the general structure of the system.

3. *Changing the approach to planning or task allocation*. Within the framework of this class, the methods of dividing a task into subtasks, scheduling calculations and assigning resources are the object of research. In general, this class makes sense to explore when it comes to automatic parallelization.

Within the framework of this review, it is proposed to pay key attention to the first class, since it is the most difficult to implement, but also the most promising. What is mentioned in modern discourse as an alternative to classical systems? As a rule, the most mentioned areas are the following:

1. *Quantum computing*
2. *Dataflow computing*

Analyzing the second class, it can be divided into two subclasses that partially overlap, but at the same time are relatively independent. These subclasses address the following issues:

1. *Network hardware and software*. This includes types of network equipment such as switches or routers, existing network protocols (Gigabit Ethernet, InfiniBand) and architectures (SDN).

2. *Structural composition*. This includes how the elements are connected in the network – that is, its topology. This includes 3D-Tor, Dragonfly, fat trees and other solutions.

As for the third class, it deals with scheduling and how a task can be automatically divided into subtasks. Often, this issue depends on the programmer or the OS, but in planning, 2 key approaches can be distinguished: static and dynamic. Since this question depends, not least, on the architecture, it makes sense to postpone its review until the review of the key decisions of the previous classes.

Quantum computing

Quantum computing is a very promising field of computer science, to which a large number of works are devoted. Although the practical use of modern quantum computers is limited by their high cost and small number of qubits, a number of areas have been highlighted in which they can be effectively applied, such as cryptography [1], financial analytics [2], scientific computing [3]

There are a number of approaches to implementing quantum computers, but the most popular one is based on the idea of a quantum circuit and the concept of a qubit. A qubit is a quantum unit of information that can enter a state of superposition between two binary states 0 and 1. As a result, at the time of calculations, the quantum register is simultaneously in all possible states, but with different probabilities. The quantum algorithm changes this probability to bring the quantum register closer to the response state, after which a readout is performed and decoherence occurs into one of the possible states [4].

This approach allows you to perform calculations not on one, but on all possible values of a variable, which makes it possible to get acceleration in a number of problems, thereby solving computational problems that cannot be solved on classical systems in a reasonable time. This property is called quantum advantage. It has been proven that certain minimum hardware characteristics of the system must be ensured in order to demonstrate quantum superiority, namely: a dimension of 49 qubits, a circuit depth of 40 and a two-qubit error of no more than 0.5% [5]. At the moment, there are a number of systems that meet these requirements. For example, Google's 54-qubit Sycamore processor, which can complete a task in 200 seconds that would take a classic supercomputer 10,000 years [6]. Or the Chinese quantum photonic computer Jiuzhang, which is the second in the world to achieve quantum supremacy [7].

The described systems are full-fledged quantum computers of general purpose, but in the context of the mentioned problems, they have a lot of shortcomings. Their key limitation is the price: they are too expensive. In addition, this is not a mass-produced product that can be purchased just like that: each of these systems is a separate artificial product under the control of the developing organization/country, and therefore it is still too early to talk about the prospects for their commercial use. However, it is worth considering that the development of quantum computers does not stand still. For example, research is being conducted to speed up the development of new quantum systems. For example, Fig. 1 shows a diagram of such a full-stack development of a quantum computer based on transmons.

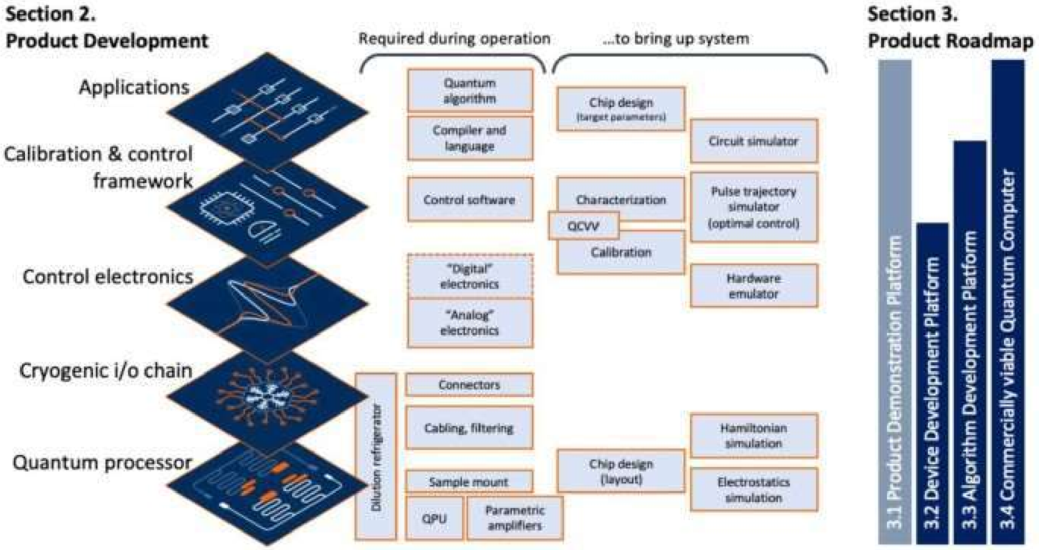


Fig. 1. Scheme of full-stack development of a quantum computer based on transmons [8]

This allows us to say that although modern quantum systems are not a potential solution to the problem of efficiency, the emergence of new, more advanced systems can fundamentally change this situation. Thus, it is necessary to take this fact into account and, considering other architectural solutions, leave room for heterogeneity and quantum integration.

However, such quantum systems are not the only solution. An alternative implementation is presented by the D-Wave company, whose processors have reached a size of 1000 qubits and continue to grow. This became possible thanks to the use of quantum annealing [9]. This method, focused on finding the global minimum of the function, is based on the fact that at the beginning of the calculation, all states of the register have equal probability, but in the process of quantum evolution, there is an ascent to a state in which the energy is minimal. This allows the processor to effectively solve optimization problems and significantly reduces the complexity of scaling, but makes it unsuitable for the execution of ordinary quantum algorithms. On the other hand, within the proposed class of tasks, this type of processor demonstrates extremely high efficiency, besides, D-Wave processors, although they have a considerable price, are available for commercial use. Therefore, it makes sense to consider their architecture in more detail.

In fig. 2 presents the architecture of one of the latest D-Wave Advantage quantum computers, which consists of several layers: an application layer, an Ocean software layer that deals with compilation and assignment of tasks, and a computer resource layer that contains 2 types of processors: classical CPUs and quantum QPUs.

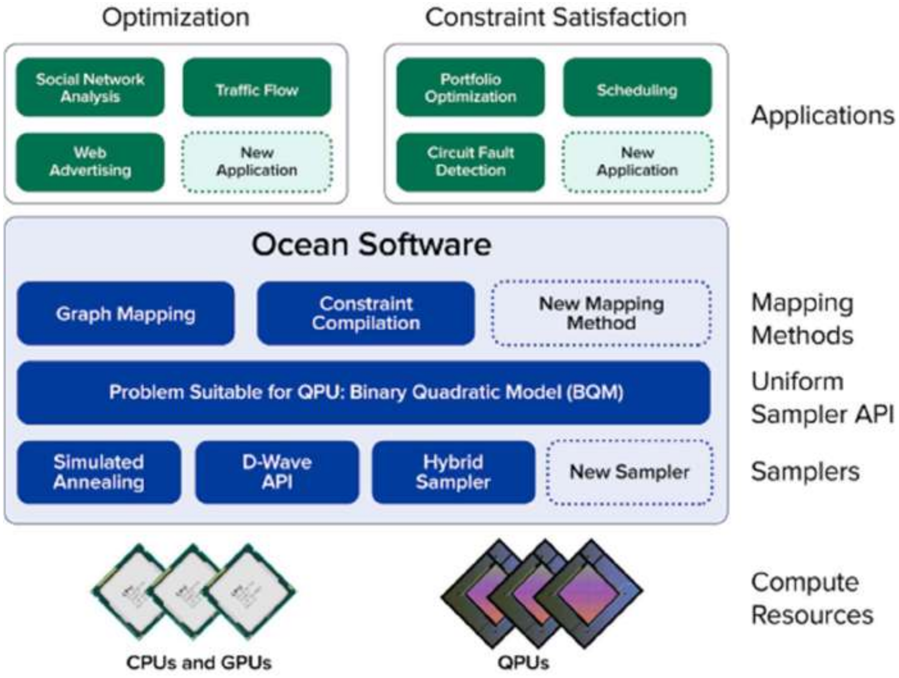


Fig. 2. D-Wave Advantage architecture [10]

Thus, D-Wave's approach allows you to make the system heterogeneous from the very beginning, combining fast, but task-limited QPUs with general-purpose CPUs. This makes it possible to almost completely eliminate any problems related to specialization, as well as to ensure efficient parallelization of calculations.

However, the question arises: how are the QPUs themselves arranged? Just as a classical system consists of nodes connected by a network, a quantum system consists of qubits connected by a graph of possible quantum entangled pairs. In the process of executing a quantum algorithm, the program connects qubits into a general quantum system, using certain connections from those available on the graph, and then performs certain transformations on the system. Thus, the characteristics of a graph are extremely important for a quantum system, because it depends on them which quantum algorithms it can perform and how quickly.

As an example, it makes sense to consider the Chimera graph presented in Fig. 3. This graph includes qubits represented in the form of horizontal and vertical loops that form a lattice. The intersection points of these loops inside each lattice implement internal connections, external connectors connect qubits in a single row or column.

However, newer systems such as the D-Wave Advantage use a slightly different graph called Pegasus (Fig. 4). It differs in that the lattice in it has an offset, which allows to expand the system of connections. It is assumed that the new D-Wave processors will use it (as well as the Zephyr graph) and not the classic Chimera graph. This graph has 3 types of connections: internal (green vertical line) connecting pairs of orthogonal horizontally oriented qubits; external (blue line) that connect adjacent vertical qubits; odd (red line) connecting equally aligned pairs of qubits.

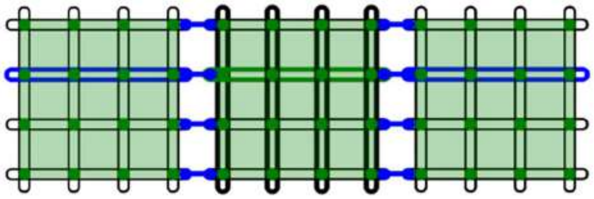


Fig. 3. Chimera graph [11]

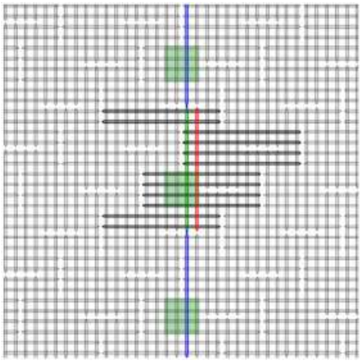


Fig. 4. Pegasus graph [11]

If you bring it to a more familiar form, with nodes and communication lines, you can get the system shown in Fig. 5. In it, green dots represent elements, green lines – connections, gray lines – splitters.

However, it is assumed that the company's future processors may switch to other topological organizations – for example, the Zephyr graph, which is presented in Fig. 6. The feature of this graph is the degree of 20 and the nominal length of 16, which potentially makes it possible to implement more qubits on a chip, as well as connect them more efficiently.

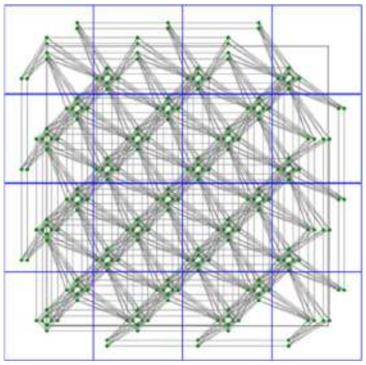


Fig. 5. Pegasus graph – topo representation [11]

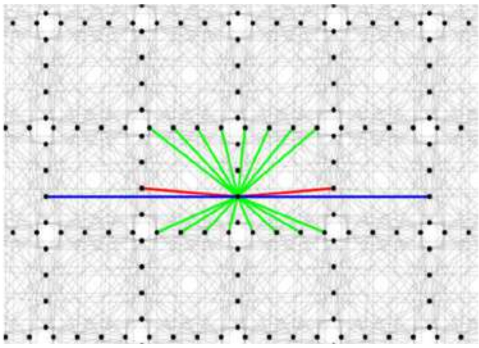


Fig. 6. Zephyr graph [11]

In the context of the problems of the subject area, quantum computing is interesting for the reason that quantum parallelism is an almost ideal form of parallelism as such. The calculation is always performed simultaneously over the entire field of solutions, independently processing each of them, and the subject of processing is not only the values themselves, but the probabilities of obtaining them when reading. This makes it possible to almost completely eliminate data dependencies from the problem, which is one of the key factors of slowdown. Also, quantum problems do not know the inconsistency of the algorithm and the task or balance problems: yes, for each task, the amount of "quantum flows" that it needs is allocated, and it is usually simply impossible to perform the task with a smaller number of qubits than it needs. The key limitations of quantum computing are its cost and the need to develop special quantum software to gain real benefits.

Dataflow as alternative paradigm.

In terms of computing management, 2 key paradigms are distinguished: contolflow and dataflow. The first approach is classic: the system has a command counter, and the program is presented in the form of a strict sequence of instructions. Accordingly, the parallel contolflow system contains a number of devices that have their own counters, and the control relies on processes and flows – a sequence of commands that is the result of breaking down the original task. However, from the point of view of parallelism, this model is not convenient: it loses internal parallelism, additional dependencies and idles appear, and when multitasking – context switching.

An alternative is the dataflow approach, in which the task is presented in the form of computational tasks connected by data. Execution occurs when ready: ready-made tasks can be executed when a free resource is available, regardless of the sequence in which they are described in the program.

Historically, the first dataflow machines implemented parallelism at the command level. However, this approach turned out to be ineffective due to high overhead costs. Therefore, a number of concepts were formed, which can be conventionally divided into 3 ways. The first path, which can be conventionally called hybridization, gave rise to modern RISC processors and superscalar architecture [12]. The idea of this approach is to hide the dataflow component inside another architecture, thereby using common tools at the programming level, and performing dependency detection at the command level, thereby speeding up the execution of independent parts.

The second way, emulation, is based on describing the tasks and data relationships at the programming level and implementing a parallel algorithm so that its parallel parts run when ready, regardless of how they are actually written in the program. Hardware-wise, the system remains pure contolflow. This includes the dataflow approach to programming and modern parallelizing compilers that simulate execution in a dataflow system, and then form a parallel algorithm from a sequential one based on the model [13, 14].

The third way to solve the problem is the development of the architecture itself: streaming, coarse-grained, vector dataflow machines, dataflow networks, and FPGA-based devices oriented to specific tasks. Within the framework of the described problems, it makes sense to focus on them and their hardware components.

Threaded dataflow and coarse-grain solutions extend the classic fine-grained parallelism of dataflow with the possibility of parallelism at the level of threads and subtasks – that is, medium and coarse-grained. This approach allows to reduce dynamics overhead due to the fact that the execution time of each specific task becomes much greater than the time of hardware planning of calculations, which eliminates delays in the execution of individual commands and reduces the need for associative memory. Sometimes dataflow modules in such systems are offered as an add-on to conventional von Neumann processors used as processing elements, but this approach is criticized for the fact that such a system by definition cannot work faster than classic controlflow. An alternative here is the use of simpler elements.

The approach of vector dataflows is similar, but slightly different: instead of increasing a single grain (and increasing the number of operations at the same time as the processing time increases), they propose to increase the volume of the same type of calculations performed simultaneously, due to vectorization. So, in this type of system, the commands are not scalar, but vector, which allows you to ignore the planning time due to the general acceleration of calculations from vectorization.

The third hardware approach, which is extremely popular today, is the use of FPGAs as the hardware base for the system [15, 16]. This approach is based on several things: firstly, the dataflow system is developed here for a certain task, which allows you to optimize it and, if necessary, include additional architectural elements – pipeline, vectorization, matrix or coarse-grained approach. Secondly, due to the manipulation of the number of devices, a certain balance between energy consumption and speed is achieved, which is especially relevant for embedded devices [15].

In the context of modern high-performance dataflow computing, the best known is the dataflow concept of the Maxeler company, which uses FPGA-based accelerators and the parallelizing compiler MaxCompiler, which turns a sequential controlflow program into a parallel dataflow application. This approach solves a number of applied issues at once: on the one hand, it harmonizes new solutions with previously created hardware and software, and on the other hand, it allows you to get a real acceleration and make the system more efficient. Therefore, it makes sense to pay attention to him.

How does this system work? Fig. 7 shows the main stages of its work. First, the Java program is compiled into a .max file of parallel tasks. It then passes through Maxeler's operating system and system software, where some of the computing scheduling tasks are performed. In the next step, the scheduled computing tasks are submitted to the hardware dataflow accelerator (DFE).

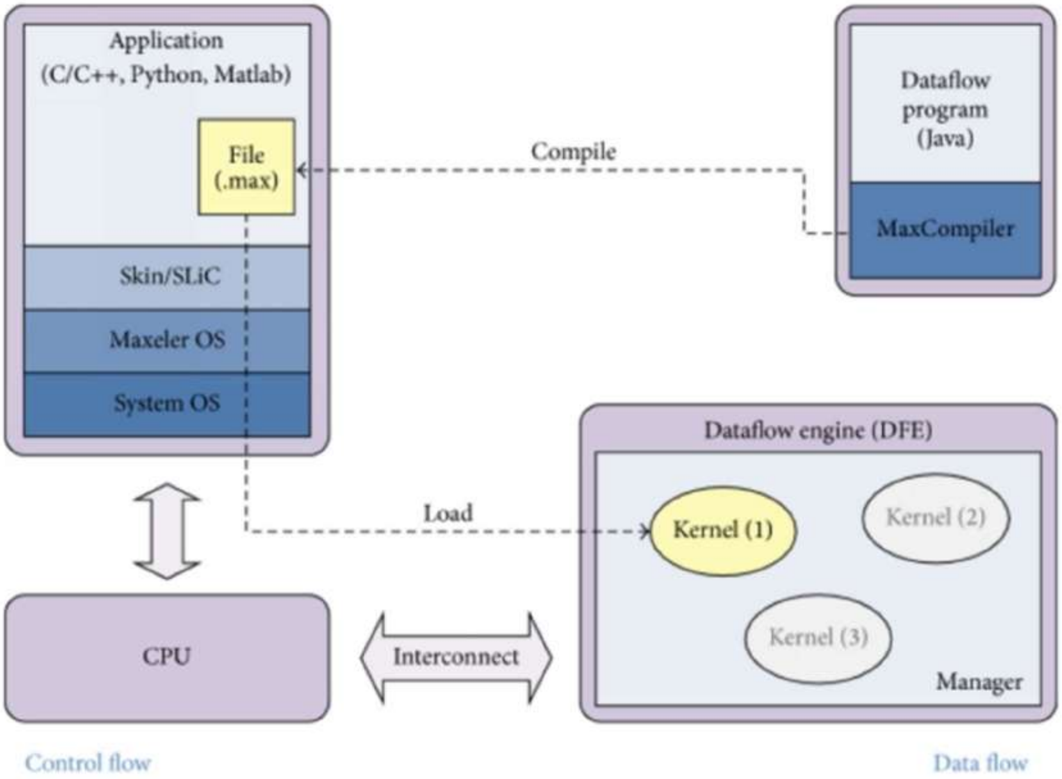


Fig. 7. Maxeler MPCarchitecture [17]

However, this is a general MPC architecture. To better understand how this should work in high-performance computing, it is worth diving deeper into the solutions offered by the company. At the moment, 4 architectures are offered for cluster and network computing (MPC-X, MPC-C, MPC-N, JDFE), the MaxCloud infrastructure, which provides dataflow computing services in the cloud, as well as the Desktop version of the accelerator.

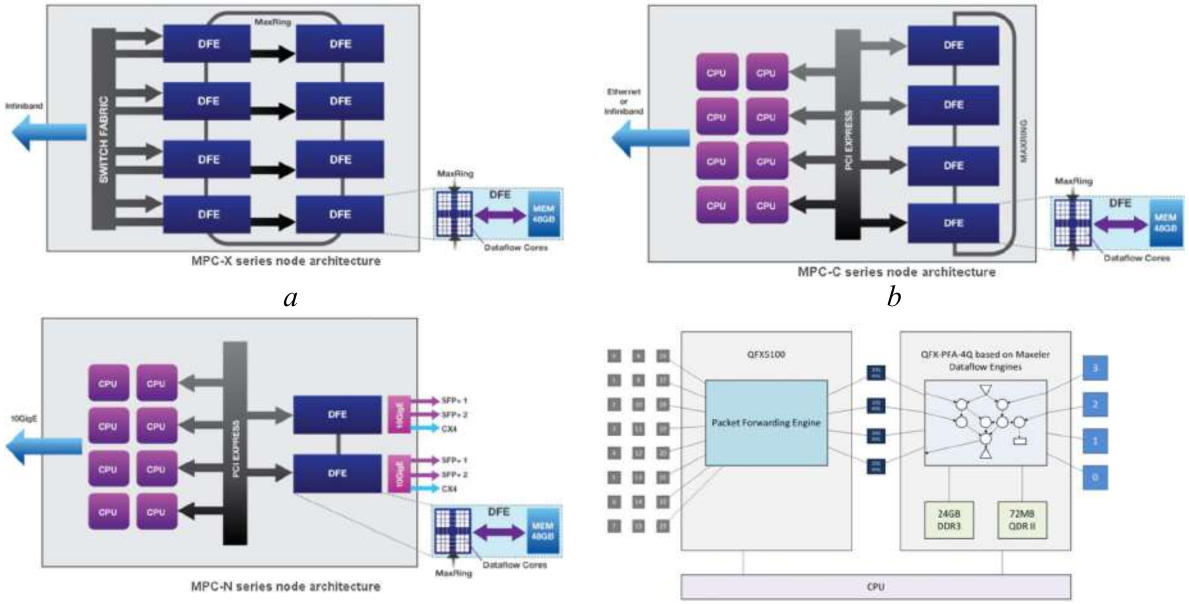
You should start with the MPC-X series. It is focused on the maximum use of dataflow calculations and in the latest MPC-X2000 implementation contains 8 MAX4 (Maia) processors in each node, connected inside the node using MaxRing and the InfiniBand network at the general level. Thus, individual MPC-X2000 nodes contain up to 768 GB of memory and provide more than ten times the acceleration compared to classic x86 servers [18].

A different approach is offered in the MPC-C series. Unlike MPC-X, it is a hybrid, combining CPU and DFE. Each MPC-C500 node contains 4 Vectis DFEs and 12 Intel Xeon CPUs, and provides a total of up to 192 GB of internal memory per CPU and per DFE (up to 384 GB of memory per node). InfiniBand or Ethernet can act as a common network here [19].

The MPC-N series is focused on minimizing delays and fast processing of data streams with a transmission speed of up to 10 Gb/s. In addition to 2 DFE Vectis and 12 Intel Xeon, its nodes contain 4 SPF/SPF+ ports, 2 CX4 ports, and also provide additional synchronization capabilities and network protocol support [20].

The latest series, JDFE aims to combine software-defined networking (SDN) technology and dataflow computing, separating the control plane and the data plane and enabling data plane programming via Maxeler Dataflow. Such a system contains thousands of small dataflow cores, using massive parallelism of calculations and providing almost a 100-fold advantage in speed while maintaining the size and power consumption at the level of a classical system [21].

In fig. 8 shows the architecture of the main 4 series of the Maxeler company, a brief overview of which is given above.



a *b*

c *d*

Fig. 8. Maxeler's main architectural solutions are: *a* – MPC-X series architecture, *b* – MPC-C series architecture, *c* – MPC-N series architecture, *d* – JDFE series architecture

In the context of the described issues, dataflow as a paradigm has 2 main advantages. First, it provides a much higher speed of operation with the same dimensions and the amount of energy consumption of the system. Secondly, the implementation of the dataflow system does not require a non-classical element base (such as qubits), and the presence of programmable FPGA logic circuits allows you to avoid problems associated with the deployment of mass production. A partial drawback is the specificity of dataflow programming, but as Maxeler's experience shows, this problem is not insurmountable.

Network level and its specifics

There are a significant number of aspects in this subject area. Conventionally, they can be divided into hardware and topological, but such a division will not be completely accurate, since the hardware is strongly connected with the protocols, and the protocols – with the topology. Thus, some solutions are complex, while others involve a certain degree of freedom. Therefore, it makes sense to expand the classification:

1. *Hardware level solution*. This includes hardware software or hardware software complexes that offer only hardware tools and common communication protocols, but do not determine the topological level of the network. For example, InfiniBand or SDN.
2. *Structural level solutions*. This includes structural compositions that determine the relationship of elements and the specifics of routing, but do not determine the equipment that the network should consist of. This includes topologies such as 3D-Tor, hypercube or fat tree.
3. *Communication protocols*. This includes protocol solutions and algorithms that do not determine either the hardware or the topological organization of the network. For example, tabular routing protocols.
4. *Combined solutions*. This includes solutions in which hardware, protocol and topology are an integral whole. This includes, for example, the TokenRing protocol, which imposes certain requirements on the equipment and is oriented towards the ring topology.

Analyzing the given classification, one can see some specifics. Yes, hardware-level analysis is key, but it depends on the architecture-level decision, so it must be analyzed separately. Similarly, this applies to the combined approach. Protocol decisions are by definition quite abstract, but they should be chosen based on the needs of the system, which makes general analysis meaningless and even harmful. Thus, the question arises about the expediency of structural level analysis. This level is also abstract, but any network needs a topology, and the issue of routing and data transmission is relevant regardless of the specific architectural approach implemented at the node level. Thus, the current analysis of network solutions should be considered precisely in the context of topology.

As a result, the question arises: what topologies are used in modern systems? At the moment, Fat trees and Dragonfly are quite popular for high-performance systems and network data centers [22].

The idea of a fat tree is as follows: there are elements located in the leaves of the tree, and there are switches that make up the main part of the tree. The closer the switch is to the root, the better the bandwidth parameters it has. In fig. 9 presents the structure of this topological organization, which consists of 4 levels.

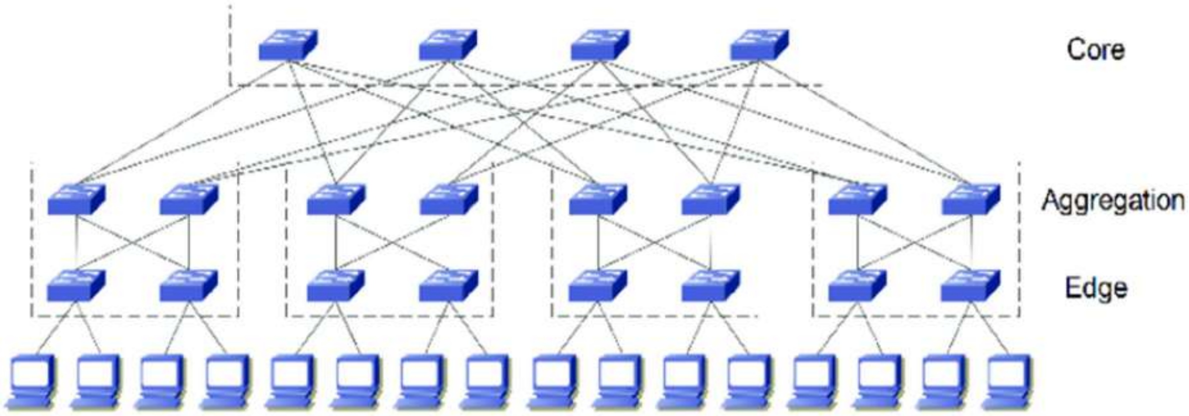


Fig. 9. Fat Tree topology [23]

What does it do in terms of performance? First, since the topology has redundant (compared to a regular tree) connections, it allows you to speed up routing and make it more reliable. Moreover, thanks to the use of multipath routing methods, it becomes possible to simultaneously transmit information through parallel channels, which are quite numerous in this topology. An equally important property of a fatty tree is its variability. Yes, there are a large number of ways to configure the topology depending on the needs and available hardware. Thus, this structural organization allows solving the main problems of supercomputers and data centers, which makes it popular. Yes, it is used in Summit, Sierra supercomputers, as well as other high-performance systems [24].

Another solution is offered by the Dragonfly topology. Its idea is fractality: yes, the system is divided into groups connected by a common network, while the groups consist of routers to which the nodes are connected. As a result, at each of the levels, the diameter of the topology is equal to (or close to) 1, and the number of nodes with scaling grows quite quickly. In fig. 10 presents the structure of the topology.

Like a fat tree, Dragonfly is a variable topology whose structure is defined by four parameters (p, a, g, h). At the same time, p is the number of terminal connections of nodes to the router, a is the number of routers in each group, g is the number of groups, and h is the number of external connections between groups. Varying these parameters allows you to change the characteristics of the network, and therefore – to select such configurations that would satisfy the purpose. In fig. 11 shows some variants of the Dragonfly topology depending on different parameters a, g and h.

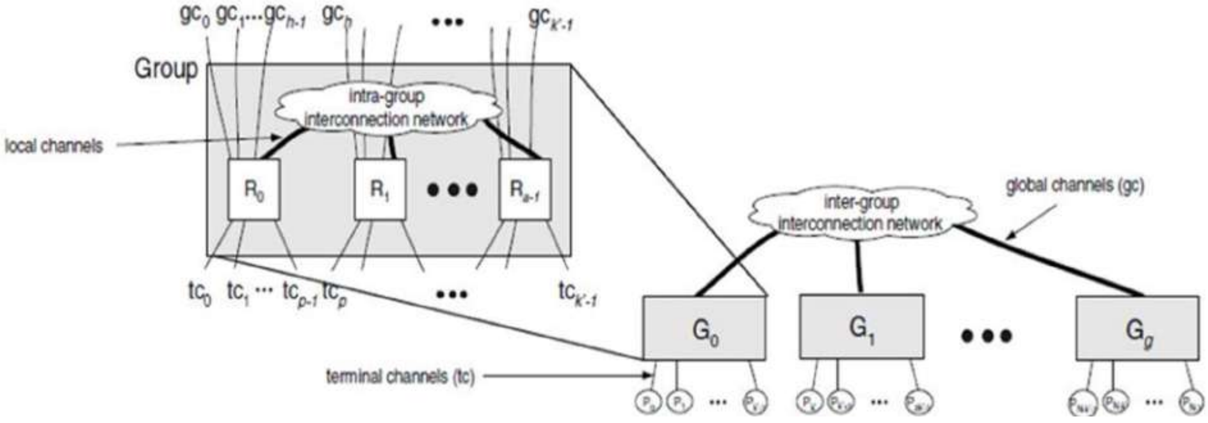
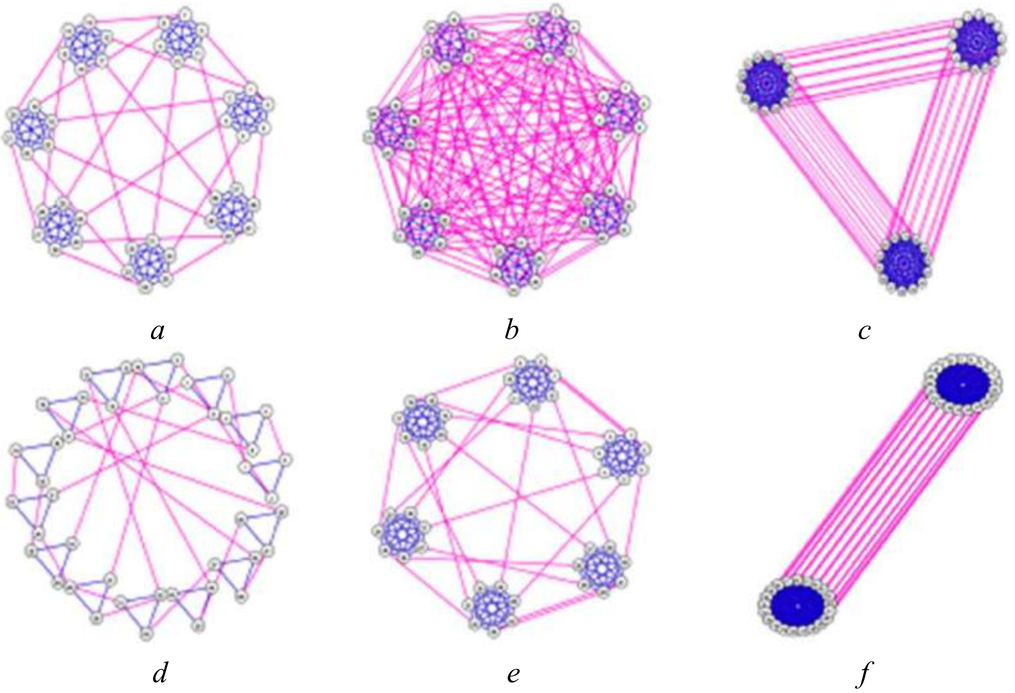


Fig. 10. Dragonfly topology – structure [25]



a *b* *c*

d *e* *f*

Fig. 11. Dragonfly topology – variants [26]: a – “Canonical” Dragonfly variant with *a* = 6, *g* = 7 and *h* = 1; *b* – Dragonfly variant with a = 6, g = 7 and *h* = 1; *c* – Dragonfly variant with a = 14, g = 3 and *h* = 2; *d* – Dragonfly variant with a = 3, g = 14 and *h* = 1; *e* – Dragonfly variant with a = 7, g = 6 and *h* = 1; *f* – Dragonfly variant with a = 21, g = 2 and *h* = 1.

Comparative analysis

Summarizing the review, it makes sense to analyze the solutions one by one. First, specific decisions within each direction. Then – the best decisions of directions within the class among themselves. Table 1 shows a comparison of classical quantum computers with specialized systems of the D-Wave company. The optimal parameters are highlighted in green.

As can be seen from the comparison, D-Wave computers (e.g., D-Wave Advantage), although limited in the class of tasks, provide approximately 16 – 50 times more resources (qubits) at a price that is 200 times lower. Also, it's worth noting that D-Wave processors are available for purchase, while general-purpose quantum systems are typically only available through a QC-as-a-service model. In the context of high-performance computing, where every clock counts, the lack of physical access to the chip is critical. This makes the choice unequivocal in favor of D-Wave systems.

Table 1.
Comparative analysis of quantum systems

Characteristic	Universal QS (quantum circuit	D-Wave QS (quantum annealing)
Types of tasks solved	All quantum algorithms with acceleration, classical algorithms with CPU speed.	Optimization problem with acceleration
Number of qubits	53 (Google Sycamore), 127 (IBM Eagle)	2048 (q2000), 5640 (Advantage)
Cost	$ 3.000.000.000 (IBM Eagle)	$ 15.000.000 (q2000)

As can be seen from the comparison, D-Wave computers (e.g., D-Wave Advantage), although limited in the class of tasks, provide approximately 16 – 50 times more resources (qubits) at a price that is 200 times lower. Also, it's worth noting that D-Wave processors are available for purchase, while general-purpose quantum systems are typically only available through a QC-as-a-service model. In the context of high-performance computing, where every clock counts, the lack of physical access to the chip is critical. This makes the choice unequivocal in favor of D-Wave systems.

Table 2 presents a comparison of solutions offered by Maxeler for dataflow computing. Since each of them has its own specific orientation, it is almost impossible to single out the best among them.

Table 2.
Comparative analysis Maxeler MPC Dataflow [18 – 20]

Characteristic	MPC-X2000	MPC-C500	MPC-N40	MPC-N42
CPU	–	12 Intel Xeon	12 Intel Xeon	
DFE	8 Maia	4 Vectis	2 Vectis	
CPU RAM	–	Up to 192 GB		
DFE RAM	96 GB per DFE	48 GB per DFE	24 GB per DFE	
Inner network	MaxRing	MaxRing (DFE), PCI Express (DFE-CPU)	PCI Express 8x2	
Global network	InfiniBand	Ethernet, InfiniBand	100GigE	
Диски	–	3 x 3.5”, 5 x 2.5”	3 x 3.5”	16 x 2.5”
Additional I/O ports	–	–	4 SFP/SFP+, 2 CX4	
Purpose	Computing	Computing, CPU-DFE hybridization	Data stream processing (10 Gbit/s) with minimal latency	

In the context of high-performance computing, the MPC-X series is the most interesting, but other types of nodes could also be useful depending on the specifics of the task and the overall network architecture.

After completing the analysis within each of the directions, it is useful to perform the same analysis between the directions within each of the classes. Table 3 compares the considered architectural directions, analyzing their suitability for solving the research problem.

As can be seen from the analysis, D-Wave quantum computers solve some problems almost perfectly, but their specialization makes them unsuitable for use instead of CPUs. Another solution is dataflow processors (DFE), which, as Maxeler's experience shows, can both replace classic CPUs and be used in cooperation. But their acceleration is limited.

Table 3.
Comparison of architectural level solutions

Problem	**D-Wave quantum computing**	**Dataflow**
The problem of parallelism	It is solved automatically in an ideal way	It is resolved automatically due to dynamics, there are overheads
Programming problem	Not solved: the system needs a special approach	There is potential for automation, there are automated means of detecting dependencies
The problem of matching the task and the system	It is partially solved, but not for all problems	Solved at the destination automation level (parallelism is exposed as fully as possible)
The problem of balancing productivity and costs	Does not occur: the problem can either be solved or not	Can be resolved through OS level or FPGA properties
Interaction problem	There is an interaction of qubits due to quantum entanglement	Available node interaction through classic data transfer

The following analysis, presented in Table 4, deals with networks, and more precisely, with network topologies. However, if architectural solutions need to be analyzed according to the problem, network solutions cannot be analyzed in a similar way, but topological characteristics of networks and their more general properties can be evaluated: for example, the availability of alternative routes, which is important in the context of multipath routing and fault tolerance.

Table 4.
Topological solutions

Characteristic	**Fat tree**	**Dragonfly**
Topological characteristics		
Degree	Depends on connectivity, but no more than 4 – 6.	p + a – 1 + h [**Помилка! Джерело посилання не знайдено.**]
Diameter	No more than a binary tree	Minimum, with full connectivity within each level – 5 (between terminal nodes)
Possibility to solution of the problem		
Alternative routes	Yes, due to additional inter-level connections	Yes, due to local full connectivity
Structure-oriented routing	Topology-based depth-first search	Based on the properties of connections
Bandwidth problem	Topology is focused on solving this problem	There are methods of topological routing that solve the problem
Optimal types of routing	Centralized	From the source or decentralized

Analyzing the topologies, one can notice certain similarities: both of them are polymorphic, have good characteristics and implement multipath routing. On the other hand, a fat tree relies heavily on centralized routing, where route discovery is performed by a shared controller, allowing for consideration of all nuances with increased bandwidth near the roots. On the other hand, the Dragonfly offers greater decentralization and a potentially better diameter, making it more promising in terms of efficiency.

Discussions

Analyzing the given solutions, it is possible to draw a conclusion about the possibility of their combination. From the point of view of the dataflow network, the system is not much different from the classic controlflow, which allows you to use a network of any hardware type (InfiniBand, Ethernet, SDN) and any topology. At the same time, quantum systems are specific in this aspect. Combining qubits requires special graphs, as the overview of D-Wave graphs demonstrates, but hypothetically the network between QPUs operates on classical data and therefore can have any structure.

However, a much more interesting combination is the combination of two architectures: dataflow and quantum processors. The first architecture significantly increases the efficiency of calculations, but its acceleration is limited. The second – makes it possible to significantly simplify a number of problems due to quantum advantage, but is specialized. Although at the moment quantum chips are too expensive, there is a significant possibility that with the development of technologies, their price will decrease further, and therefore, at a certain point, it will be possible to use them in part of the nodes of the system as accelerators. As a result, the only question is in which specific parts of the system these nodes should be located. The topology comes in handy here, the right choice of which will allow for effective access to the quantum resource.

Conclusions

Summarizing the review, the following aspects should be noted. The first is that the problem of limited efficiency is complex and consists of a number of smaller problems. There is a large number of possible solutions for each of them, which makes their classification relevant. This article proposes to divide them into those related to the immediate structure of the node (architectural level) and those related to the interaction of nodes and abstracted from their architecture (network level), however, this classification is not final and can be expanded.

The second aspect is that there are a large number of possible solutions for each problem, both mentioned and not mentioned in this review. All of them have their own specifics and can be both compatible with each other and contradict each other. However, since each of the proposed approaches affects only part of the problem and does not solve the problem in general, there is a need to find methods that would allow combining partial solutions into a general one.

Returning to the material of the article, the reviewed solutions allow solving certain tasks: solving the problem of parallelism, partially – simplifying programming, adapting the task to the system and the system to the task, as well as solving interaction problems. Benchmarking shows that key solutions such as the dataflow paradigm and quantum computing complement each other and can therefore be combined for better results.

However, there are a number of unsolved tasks, such as the simultaneous use of QPU and DFE, the combination of quantum and classical algorithms within the program, the planning of calculations at the network and system-wide levels. Also, the question of the relationship between architecture and structure, as well as the search for optimal hardware and topological solutions at the network level, remains unresolved. Similarly, it makes sense to expand and deepen the review, including new modern architectures and paradigms and considering the issue of hybridization as a method of solving the given problem.

References

[1] V. Mavroeidis, K. Vishi, M. D., and A. Jøsang, “The Impact of Quantum Computing on Present Cryptography,” *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 3, 2018.

[2] R. Orús, S. Mugel, and E. Lizaso, “Quantum computing for finance: Overview and prospects,” *Reviews in Physics*, vol. 4, p. 100028, 2019, doi: https://doi.org/10.1016/j.revip.2019.100028.

[3] Y. Cao *et al.*, “Quantum Chemistry in the Age of Quantum Computing,” *Chemical Reviews*, vol. 119, no. 19, pp. 10856–10915, Oct. 2019, doi: https://doi.org/10.1021/acs.chemrev.8b00803.

[4] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information*. Cambridge: University Press, 2000.

[5] J. Kelly, “A Preview of Bristlecone, Google’s New Quantum Processor,” *blog. research. google*, Mar. 05, 2018. https://blog.research.google/2018/03/a-preview-of-bristlecone-googles-new.html

[6] F. Arute *et al.*, “Quantum supremacy using a programmable superconducting processor,” *Nature*, vol. 574, no. 7779, pp. 505–510, 2019, doi: https://doi.org/10.1038/s4158601916665.

[7] P. Ball, “Physicists in China challenge Google’s ‘quantum advantage,’” *Nature*, vol. 588, no. 7838, pp. 380–380, Dec. 2020, doi: https://doi.org/10.1038/d41586-020-03434-7.

[8] Alberts, *et al.*, “Accelerating quantum computer developments,” *EPJ Quantum Technology*, vol. 8, no. 1, p. 18, 2021, doi: https://doi.org/10.1140/epjqt/s4050702100107w.

[9] C. C. McGeoch, *Adiabatic Quantum Computation and Quantum Annealing*, 1st ed. Springer Cham, 2014, pp. 1–93. doi: https://doi.org/10.1007/978-3-031-02518-1.

[10] “Quantum Computing,” *D-Wave Government*. https://dwavefederal.com/system/

[11] “D-Wave QPU Architecture: Topologies – D-Wave System Documentation documentation,” *docs.dwavesys.com*. https://docs.dwavesys.com/docs/latest/c_gs_4.html

[12] Arvind and S. Brobst, “The evolution of dataflow architectures: from static dataflow to P-RISC,” *International Journal of High Speed Computing*, vol. 05, no. 02, pp. 125–153, Jun. 1993.

[13] M. Carkci, *Dataflow and Reactive Programming Systems*. Createspace Independent Publishing Platform, 2014.

[14] O. Pochayevets, “BMDFM: a hybrid dataflow runtime parallelization environment for shared memory multiprocessors,” 2006. Available: https://mediatum.ub.tum.de/doc/601795/601795.pdf

[15] G. Gobieski, A. Nagi, N. Serafin, Isgenc, Mehmet Meric, N. Beckmann, and B. Lucia, “MANIC: A vector-dataflow architecture for ultra-low-power embedded systems,” Columbus, OH, USA: Association for Computing Machinery, 2019, pp. 670–684.

[16] V. Milutinović, M. Kotlar, M. Stojanović, I. Dundic, N. Trifunović, and Z. Babović, *DataFlow Supercomputing Essentials*, 1st ed. Springer Cham, 2017.

[17] A. Kos, S. Tomažič, J. Salom, N. Trifunovic, M. Valero, and V. Milutinovic, “New Benchmarking Methodology and Programming Model for Big Data Processing,” *International Journal of Distributed Sensor Networks*, vol. 11, no. 8, p. 271752, Aug. 2015.

[18] “MPC-X Series | Maxeler Technologies,” *www.maxeler.com*. https://www.maxeler.com/ products/ mpc-xseries/

[19] “MPC-C Series | Maxeler Technologies,” *www.maxeler.com*. https://www.maxeler.com/ products/mpc-cseries/

[20] “MPC-N Series | Maxeler Technologies,” *www.maxeler.com*. https://www.maxeler.com/ products/mpc-nseries/

[21] “JDFE | Maxeler Technologies,” *www.maxeler.com*. https://www.maxeler.com/products/jdfe/

[22] M. A. Dibrova, A. V. Kogan, and A. L. Vorobyova, “Method of forming a plurality of paths in the network data center,” *Bulletin of NTUU “KPI”. Informatics, control and computer engineering*, no. 63, 2015.

[23] T. Wang, Z. Su, Y. Xia, and M. Hamdi, “Rethinking the Data Center Networking: Architecture, Network Protocols, and Resource Sharing,” *IEEE Access*, vol. 2, pp. 1481–1496, 2014.

[24] N. Jain *et al.*, “Predicting the Performance Impact of Different FatTree Configurations,” in *SC17: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–13.

[25] J. Kim, W. J. Dally, S. Scott, and D. Abts, “Costefficient dragonfly topology for largescale systems,” in *2009 Conference on Optical Fiber Communication*, pp. 1–3.

[26] M. Y. Teh, J. J. Wilke, K. Bergman, and S. Rumley, “Design Space Exploration of the Dragonfly Topology,” in *High Performance Computing*, J. M. Kunkel, R. Yokota, M. Taufer, and J. Shalf, Eds., Cham: Springer International Publishing, 2017, pp. 57–74.

UDC 004.056.5 *DOI: https://doi.org/10.20535/2708-4930.3.2022.266391*

MODERN INFORMATION SYSTEMS SECURITY MEANS

A. I. Verner, I. A. Klymenko

The article provides a thorough review of current research on information security means available for the endpoint protection. In the first chapter, categories and features of types of threats to information security are considered. The second chapter provides a general description of threat analysis methods, compares static, dynamic, hybrid malware analysis methods and highlights the advantages and disadvantages of each of them. The third chapter considers the modern methods of detecting and mitigating threats to information systems, as well as the peculiarities of their implementation. The purpose of this article is to provide a general overview of the current state of information security and existing modern methods of protecting information systems from possible threats.

Key words: *information security, information systems, security means, malware.*

Introduction

High rates of technological progress and information technologies dissemination are ubiquitous nowadays. Statistical data [1] indicate that there is a concurrent pattern of yearly exponential growth in the volume of harmful software affecting information systems. This necessitates the creation of a variety of novel, adaptable forms of protection mean. Nevertheless, despite the coordinated efforts of experts, the problem of malware analysis and detection is still unresolved.

As of September 2022, research on operating system (OS) use [2] reveals that the commercial Windows OS continues to be the most popular among desktop computer users. Regarding the threats, despite the Q4 2021 Internet Security Report's conclusions that attacks were decreasing downward year over year, a large increase in threats detections in Q1 2022 indicated that the situation became worse [3]. In Q2 2022, 55,314,176 malicious and potentially unwanted objects were detected by security systems [4].

Additionally, the predominance of embedded systems, which are mostly employed in the so-called "Internet of Things" (IoT), is growing quickly. This has caused some obvious shifts in the landscape of malicious software. Due to the high rates of product release, corporations pay insufficient attention to the issue of product security, which leads to the presence of a significant number of major vulnerabilities in such systems being rather often detected. Architecturally embedded systems have strong differences from desktop personal computers, which is caused, first of all, by the use of various processors and rather limited resources. From the point of view of the operating systems usage here, of course, the situation is also radically different, since according to [5] developers use Unix-like OSes with various variations of the Linux kernel. According to evidence [6], almost half of smart homes with built-in systems had critical vulnerabilities that allowed attackers to easily attack them. The report [4] shows, that most of the IoT devices were attacked using the Telnet protocol, as before (Telnet – 82,93%, SSH – 17,07%). In addition, according to this source, there is a 217% increase in the number of attacks compared to previous years.

By analyzing the aforementioned facts, we can draw the insight that creating protection means to increase the security of information systems is a very critical issue in the contemporary.

This paper describes the actual status of information security, categorizes and highlights the specific attributes of security mechanisms depending on attacks, and examines contemporary methods for detecting threats to information systems.

Information security threats: categories and specifics

A threat in the context of information security is a potential negative action or occurrence facilitated by a vulnerability, leading to an unintended effect on a computer system or application.

Threats to information security can take a variety of forms including software attacks, intellectual property theft, identity theft, equipment theft, information theft, sabotage, and information extortion. Any software that has the potential to compromise the integrity of an information system is considered malware [7]. Malware is an acronym for malicious software and, therefore, it is essentially defined as harmful software that can be invasive computer code or anything else created with the intention of harming a system. Due to the presence of many malicious software and a huge range of programs, each type of malware can be unambiguously divided into classes. Most time it is incorrectly interpreted that, viruses, worms, and bots are all the same things. The only common feature is that they are all related to the malicious programs, however they behave in the most distinct way. As was mentioned, malware includes viruses, worms, Trojan horses, rootkits, spyware, keyloggers, etc. According to the report [8], most spread were the heuristic malware in 2021. The fig. 1 represents the graph of detected malicious software for the 2021 year.

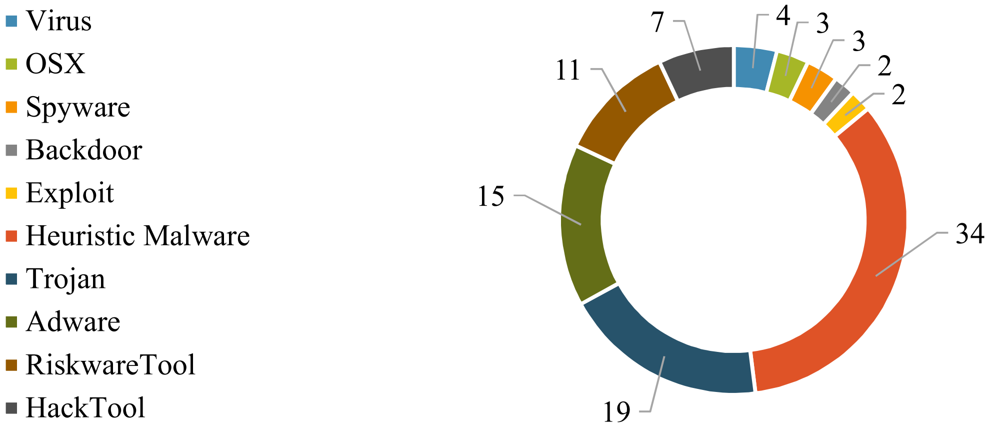


Fig. 1. Top 10 threats detection categories 2021.

The intricacies of how each form of malicious software functions will be later covered in more depth further.

Generally, the malware may be divided into two main groups by such aspects:

– methods of Infection;

– malware Behavior.

Based on the infection methods the following examples of malware can be identified:

The ***viruses*** are malicious software having a self-replication mechanism. Such software has an executable file that it uses to replicate itself to other host systems and proliferate. Because the program is passive, infection happens through files, media, or network files. The viruses could also modify its replicated copies, depending on how complicated the computer code is [9]. Viruses can be used to steal information, build botnets, show adverts, and many other activities in addition to harming computer nodes and networks.

Although ***worms*** also have a replication mechanism, they are representing an active malware that spreads over the network by taking advantage of numerous vulnerabilities in the existing software or operating system. They include malicious processes in them that may be utilized to create channels of communication and operate as active carriers. This class drastically lowers the system's performance and continuously scans its resources [10], which causes the node to become unstable and, in severe circumstances, the system to crash. Moreover, worms could produce payloads in the form of several bits of code that are created to harm the node by stealing data, erasing files, or building a bot that can connect an infected device to a botnet [9]. Unlike viruses, worms do not require human activity to spread, as they could spread and reproduce independently.

A ***Trojan*** is a piece of software that has the appearance of being trustworthy but when downloaded and run, runs any contained harmful code or files. A Trojan may have no payload or have extra malware installed in the form of viruses. Trojans, in contrast to viruses and worms, do not have a mechanism for self-replication and are only triggered when users launch them. However, the payload can include malware that enables an attacker to remotely access the computer node and carry out any nefarious deeds. The effects of Trojans programs on PCs vary depending on the extra payload and are typically enhanced by social engineering [11, 12].

A ***backdoor*** is a hidden "entrance" used to gaining access. It is occasionally made specifically by service providers as a remote tool for system checks, troubleshooting, and diagnostics. The simple existence of a backdoor is a huge security risk, as it is not difficult to detect. Attacks frequently occur as a consequence of safe backdoors, for instance with the "backdoor" virus. This type of malicious software allows for remote, illegal access to a computer system or application by taking advantage of system weaknesses and shortcomings. It operates in the background, much like any malicious software. This access provides the full range of actions to perform malicious operations on the system. Computer nodes are very vulnerable to illegal copying of files, modifications, data theft by using backdoors [13].

The ***bots*** are computer programs created to carry out particular tasks. Bots were initially created to control chat channels. While some of them are exploited for legal purposes, malicious bots are built to create botnets. A botnet is a network of node computers (zombies/bots) controlled by an attacker or botmaster. Bots infect and control another computer, which in turn infects other connected computers, forming a network of compromised computers botnet. Bots are frequently employed as spammers, for DDOS attacks, web distributors for spreading malware on file sharing, etc. The CAPTCHA tests are one tool used to defend systems against bots [11].

The behavior-based malware can be also divided in multiple parts.

A ***spyware*** is malicious software that monitors user activities by accessing operating system features. Such spyware occasionally contains extra capabilities, including the ability to impede network connections or even modify the infected system's security settings. Spread occurs by attaching to legitimate software, Trojan horses, or even through known vulnerabilities. Spyware can monitor user behavior, for example, by collecting keystrokes and sending information to a remote host to an attacker [14].

Spyware includes ***keyloggers***. Such software performs the recording in the background. The user is unaware that a recording of the keys they press on the keyboard exists. The collected data is then transmitted to the attacker over the Internet. These applications are designed to steal passwords, such as those used for online banking. They can also employ spyware to steal other types of personal data, such digital documents. Spyware and keyloggers may be downloaded to a distant node via a variety of methods. The most common is by following a link in spam e-mails or by visiting web pages designed solely to infect nodes. Even still, this kind of malware is occasionally referred to as "Trojan," as it spreads similarly to Trojans.

The ***zombies***. They function in a manner akin to spyware. The infection method is the same, except instead of spying and stealing data, they wait for a hacker's order.

An ***adware***, performs automatic display of advertisements in the form of pop-up ads on websites, etc. Most of this software is designed to assist marketers produce products that will make companies money. Some adware packages contain spyware, which might eventually have serious repercussions including tracking user activities and information theft [15].

The ***rootkits*** are the advanced and complex applications typically developed as tools to conceal regular operations on the infected node. To prevent being discovered by the system, rootkits employ a number of tools. They are extremely intrusive and challenging to get rid of because they are undetectable. They are developed with the possibility of full control over the system and obtaining the highest privileges on the infected node [11]. The majority of node protection software solutions are ineffective in identifying and removing rootkits because to their use of cloaking methods. Monitoring the computer system's activity in respect to the topic of unexpected activities, analyzing memory dumps, and scanning system file signatures are other ways to counteract them.

A ***ransomware*** infects computing nodes or a network and keeps the system locked down, demanding a ransom from users. To prevent users from accessing the infected machine, the files are often encrypted or the system is banned. Then messages appear demanding payment in order to view the data. Such malware uses the same spreading techniques as computer worms.

Although there are additional varieties of malware, these are the most well-known and widespread today. A trend towards fewer new harmful software creations and a dramatic increase in the overall quantity of malicious samples can be observed by examining the report of the past 10 years [16]. The graphs in Figure 2 and Figure 3 provide the visual representation of the problem.

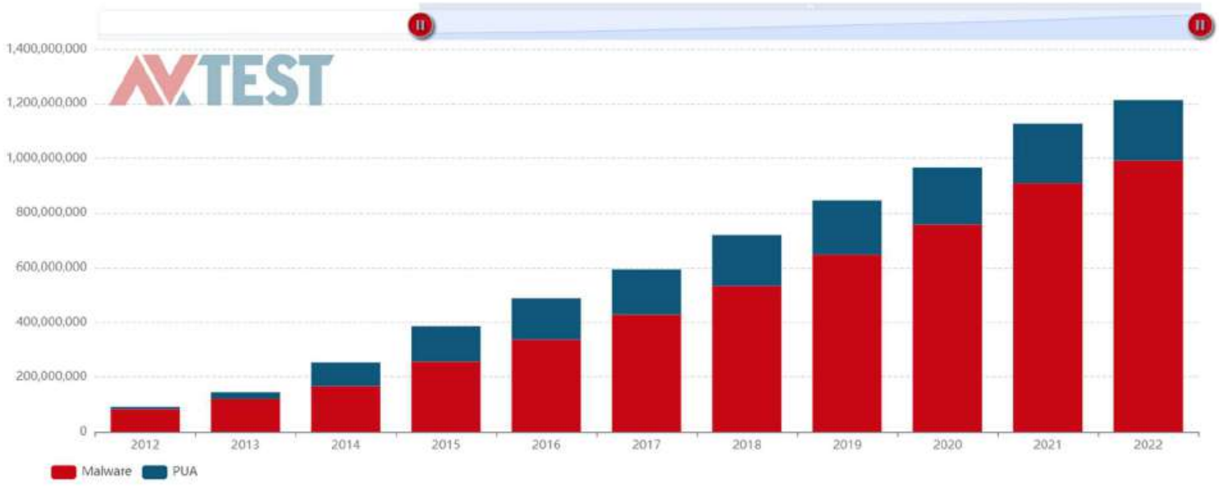


Fig. 2. Total amount of malware and potentially unwanted applications (PUA) [16].

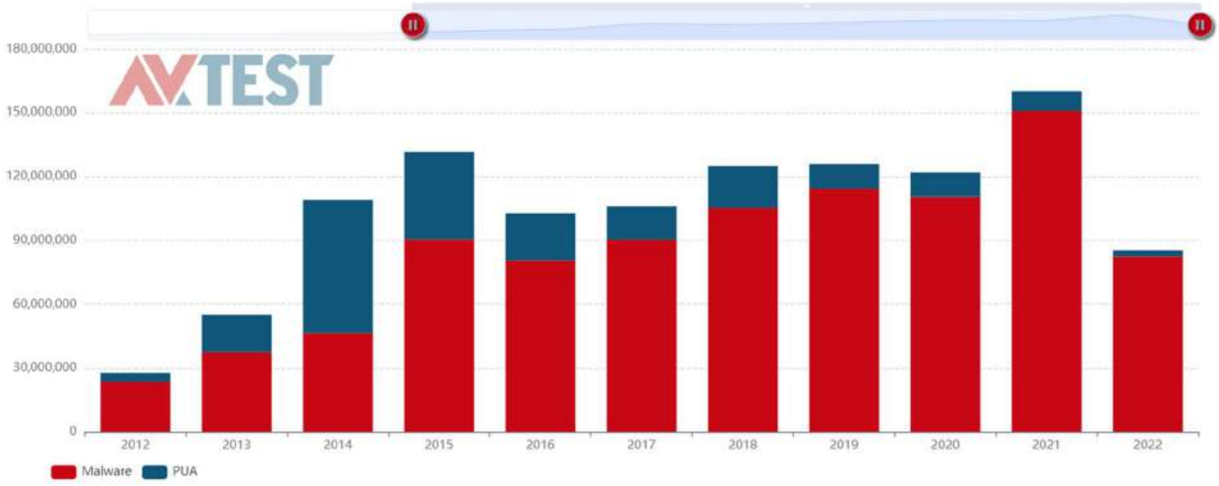


Fig. 3. The annual increase of malware and PUA [16].

From the overall evolution of new malware over the past ten years, malware prevalence is rising yearly. At the time this article was published, there were more than 1.2 billion harmful programs in use in 2022 (according to the info from av-test.org report).

The analysis and detection technologies are continually improving as a result of the fight against malware samples. Malware detection technology has been continuously evolving from rule matching and feature code extraction in the early phases to dynamic and static detection and heuristic detection in the middle phases, and finally to the current machine learning and multi-engine joint learning. Nevertheless, anti-detection technology is also improving to overcome different anti-killing methods, malware employs shell, obfuscation, virtual machine protection, and other technologies [17]. In the next chapter we will consider the modern threats detection means.

Threat analysis techniques

Malicious threats can be detected using a variety of code analysis techniques. In general, such analysis methods can be separated into three main categories: hybrid, dynamic, and static. These techniques identify and classify malicious software and take action against it in order to protect computer systems from a potential loss of data and resources.

One of the first historically developed method is the ***static analysis***. The term "static analysis" describes the examination of harmful software without actually running it. String signatures, byte sequences, *n*-grams, library syntax calls, control flow graphs, opcode frequency distribution analysis, and other detection patterns are employed during static analysis. The analysis is carried out by preliminary file's unpacking and decoding of the execution file. Debugging and memory dump analysis tools are used to reverse-engineer the basic principles of how malicious software functions. Disassemblers and debuggers allow displaying the malware code in the form of assembly instructions, which provides information about what the malware actually does, and helps identify patterns to identify attackers. Such technique is very useful for analyzing the packaged executables that are challenging to disassemble.

However, such analysis loses its effectiveness in case the obfuscation is performed. The binary obfuscation techniques convert malware binaries into self-compressed and distinctively organized files. This is generally done to prevent modification and complexifying of the overall exploration of harmful software, thus additionally reducing the opportunity of obtaining any qualitative findings. Additionally, as mentioned in [18], when binary executables are used (obtained by compiling the source code) for static analysis, details like the size of data structures or variables are lost.

Technical methods used by attackers to evade static analysis led to the development of ***dynamic analysis***. The drawbacks of the static analysis methodology were studied by Moser [2]. The scientist developed a coding obfuscation-based method that shows static analysis is inadequate for identifying or categorizing malicious software. According to the conducted studies it is confirmed that since dynamic analysis is less vulnerable to obfuscation than static analysis, it serves as an essential supplement to static analysis.

Dynamic analysis of malicious programs includes the analysis of the program during its operation in the system [19]. The malware is executed in a secure and controlled environment, to avoid the transfer of the investigated malware to other systems or networks. Observation, samples gathering and the samples interactions with the system is the foundation of dynamic analysis. For this, the snapshot of the initial state of the virtual machine is taken before the malware is launched to execute on the test system. To examine changes, the input and output states are compared. After the changes obtained from observations, they are used to further remove malware from infected nodes and/or to simulate effective signatures. Like basic static malware analysis, dynamic analysis is an important initial step in malware analysis, although it does not provide comprehensive information about the malware [20].

Extended dynamic analysis involves the use of tools to study the state of the malicious program during its execution. For instance, this allows to study the harmful code's internal state. The use of advanced analysis techniques provides information that cannot be collected using other methods [10]. Dynamic analysis is always carried out in an isolated setting to ensure that all system inputs and outputs are known for further analysis. The use of additional tools also allows to perform tracking of the APIs used at this stage, to check the system functions calls, called and deleted files, registry changes, and data processed by the program analyzed during interaction with the system. Analyzing the parameters used in API and function calls allows semantic grouping of the functions used while, analyzing the processed and distributed data in the system provides insight into the files used and produced by the malware. This allows to determine the purpose of the malicious software development [21]. The advanced dynamic malware analysis is very useful for detecting malware variants and obscured techniques. Automated dynamic malware analysis tools are employed for convenience, and they produce reports that may be utilized in order to classify harmful samples based on their behavior.

By combining both static and dynamic analysis techniques the new threat analysis approach was developed – the hybrid analysis. Such a method benefits from both approaches. A software is first examined by code analysis and malware signature validation, following which it is launched in a virtual environment to ascertain its true behavior. This allows investigating the malicious software deeply.

It is important to identify the unique peculiarities of how each type of analysis is performed:

- the static analyzers, process executables without running them and extract the classification-related information from the binaries and their metadata;
- the dynamic analysis systems execute binaries in a virtualized environment and record sample behavior, isolating the indicators of malicious activity;
- the hybrid analyzers can analyze the encrypted malware being more precise and time consuming.

While all the approaches have positives and negatives, many endpoint security solutions tend to be handled by static analyzers because of the strict time constraints required to avoid impacting system performance.

The pros and cons of using each analysis technique are briefly illustrated in the table of general approaches comparisons (Table 1).

Table 1.
Threat analysis approaches comparisons [12]

Analysis approach	Static	Dynamic	Hybrid
Pros	– Efficient. – Low influence on performance. – Safer as does not require software execution. – High accuracy.	– Has better accuracy over static analysis.	– Far superior to static and dynamic analysis. – Can detect malware that is both known and undiscovered. – Can analyze the encrypted malware
Cons	– Unknown and encrypted malware cannot be analyzed. – Unable to recognize obscure malware.	– Unsafe and time consuming – High resources utilization	– Most time and resources consuming. – Most complex

Malicious threat detection techniques

Numerous techniques for identifying threats are developed as academic study on malware detection increases. Let's examine the primary methods for detecting malicious software on computing nodes in more detail.

It is impossible to categorically say that one method is superior to another when it comes to finding significant traits because each strategy is unique and has its own benefits as well as disadvantages.

Using behavioral modeling, heuristics, and simulation-based threat detection approaches, a large amount of malware can be identified. In addition, these models also allow detecting a new species of malware. However, they are not universal and cannot detect all the malicious software developed. Therefore, there is a need to find a method that would effectively detect even more complex, still unknown programs. Overview of malware detection approaches, features, and used techniques can be seen in Figure 4.

The ***signature-based*** detection technique was initially common. A signature is a feature of the malware that encapsulates the structure of the program and identifies each malware as unique. This technique rapidly and effectively recognizes known malware species. That is why the signature detection approach widely used in commercial antivirus applications.

This approach is fast and effective enough to detect known types of malware, but not powerful enough to detect unknown types of malware. Therefore, malicious software that exploits the zero-day vulnerabilities cannot be identified by using such methodology. Additionally, by utilizing obfuscation, malware from the same species can easily avoid detection by signature-based methods [17]. As the method has such weaknesses, later other techniques emerged.

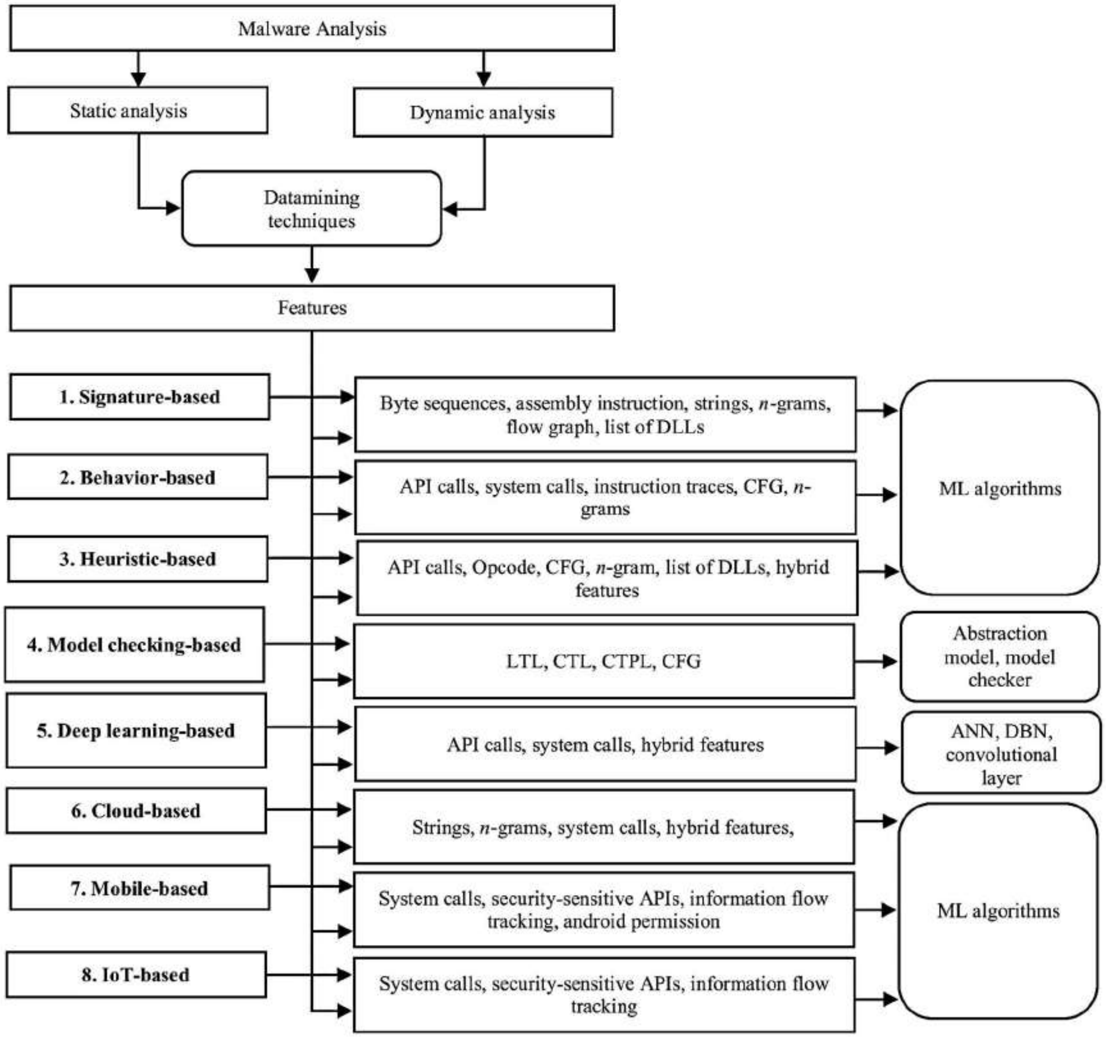

Malware Analysis
Static analysis
Dynamic analysis
Datamining techniques
Features
1. Signature-based
2. Behavior-based
3. Heuristic-based
4. Model checking-based
5. Deep learning-based
6. Cloud-based
7. Mobile-based
8. IoT-based
Byte sequences, assembly instruction, strings, n-grams, flow graph, list of DLLs
API calls, system calls, instruction traces, CFG, n-grams
API calls, Opcode, CFG, n-gram, list of DLLs, hybrid features
LTL, CTL, CTPL, CFG
API calls, system calls, hybrid features
Strings, n-grams, system calls, hybrid features,
System calls, security-sensitive APIs, information flow tracking, android permission
System calls, security-sensitive APIs, information flow tracking
ML algorithms
Abstraction model, model checker
ANN, DBN, convolutional layer
ML algorithms

Fig. 4. A diagram illustrating malware detection techniques and tools [4].

A ***behavioral method*** to malware detection uses monitoring tools to track the activity of the program and assess whether it is malicious. This method may be used to recognize the majority of new harmful software since behavior does not really change even when the software code does [21].

A malware sample could be incorrectly classified as harmless since some malware programs do not work properly in a secure environment. In behavior-based detection, features are first excluded from the dataset using data mining, and behaviors are then identified using one of the methods mentioned above. Then, using ML algorithms, particular characteristics from the dataset are retrieved and classification is performed.

A ***heuristic technique*** to malware detection has been popular in recent years. It is a sophisticated detection technique that draws on knowledge and a variety of approaches, including rules and machine learning (ML) techniques [23]. This method offers the opportunity to identify zero-day vulnerabilities, however it is unable to detect sophisticated software.

A ***model checking*** based approach. In this approach, malware behaviors are defined manually, and groups of behaviors are coded using linear temporal logic (LTL) to represent relevant features. Programmatic behavior is created by looking at the flow relationships of one or more system calls and defining the behavior using properties such as hiding, propagating, and injecting [23].

By comparing these behaviors, it is determined whether the program is malicious. This technique enables the detection of certain new software, but it cannot be used to detect a new generation of dangerous software.

Deep learning is a type of ML machine learning that inherits from artificial neural networks (ANNs) that learn from examples. This is a new approach that is widely used for image processing, drone control and voice control; however, it is still underutilized for malware detection. Although it is quite effective, its main drawback is that it is not resistant to attacks that use evasion.

Cloud computing is quickly expanding because it offers several benefits such as simple access, on-demand storage, and cost savings. Because the cloud is so widespread, it has also been used to identify viruses. With significantly larger malware databases and heavy computing resources, cloud-based malware detection improves detection performance for PCs and mobile devices.

Cloud-based detection employs several sorts of detection agents on cloud servers and provides security as a service. A user may submit any sort of file and obtain a report indicating whether the file is malware (e.g., Virus Total platform). Despite its benefits, this detection architecture has certain drawbacks.

Some drawbacks include the following:

- The cloud detection mechanism has some overhead over other detection mechanisms, so communication between the client and server must be optimized, especially for the Internet of Things and mobile devices.
- User must upload content to the cloud, which may reveal some sensitive data, such as location, password, and credit card information.
- The absence of real-time monitoring across all resources for all files.

The Internet of Things (IoT) architecture is composed of a wide range of Internet-connected smart devices such as household appliances, network cameras, and sensors. IoT and mobile devices have begun to outnumber PCs on the Internet. As mobile and IoT devices become more popular among consumers, they also become increasingly popular targets for attackers. As a result, the malware detection paradigm landscape is shifting away from desktops and toward IoT and mobile devices.

A novel approach for identifying DDoS malware in IoT contexts proposes malware categorization using convolutional neural networks and malware binary image analysis [24]. Being fast and lightweight, the mentioned method remains vulnerable to complex code obfuscation techniques. Partially this can be fixed by using static sequences and calls features limited to a certain degree.

One more method describes the detection of the cryptoransomware in IoT networks based on energy consumption footprint [25]. To accomplish malware application categorization, this technique likewise employs ML algorithms and tracks the energy consumption trends of several activities. However, the technique description proposed is unclear. Furthermore, there is no information on which ransomware family was examined or how they dealt with unknown malware.

Finally, lets briefly outline the benefits and drawbacks of each of the methods discussed above.

The signature-based approach allows performing the fast and efficient detection of known software. This method also proves its efficiency in malware detection in case the samples belong to same species. Unfortunately, such threat mean is unable to detect new types of malwares or the modification of the old one. Furthermore, it is not resistant to obfuscation and polymorphism.

The behavior-based approach has proven its validity for identifying the new malware types as it determines the malware functionality. Such method also allows detecting different species of the same malware being effective against polymorphism and obfuscation. One of the mentioned method's drawbacks is that it may produce the false positives due to the difficulty of the malicious and normal behavior separation.

Unlike above-mentioned approaches, the heuristic-based method allows detecting the unknown malware by using the combination of the static and dynamic analysis features. However, this way is a bit complex as it contains various number of rules and training phases being vulnerable to metamorphic techniques.

The model checking-based approach is complex and resource-intensive technique. However, it allows detecting the malware from the same family and is resistant to the polymorphism and obfuscation techniques.

One of the most powerful and effective is the deep learning-based approach. This method consumes some time during the detection and is not resistant to evasion attacks.

To enhance the detection performance for PCs the Cloud-based solution can be used. It provides better computational resources and bigger malware databases. Additionally, it can be easily accessed, managed and updated. However, as cloud is the remote source, some sensitive data leaks are also possible. Additionally, it requires continuous connection between the client and the server.

The last approach becomes more common nowadays due to the wide spread of the IoT devices. This approach similarly to the previous allows using both the static and dynamic analysis feature being limited to the uncomplex malware only.

Conclusions

Although the new approaches for the security means are being developed and enhanced daily, there is a still strong need in the development of the threat detection methods due to the prevalence of the malicious software nowadays. The article provided a thorough review of current research for malware detection methodologies, as well as techniques and algorithms utilized for malware detection. The benefits and drawbacks of each malware detection method have been discussed.

The most significant disadvantage of current security measures is their sensitivity to obfuscation. The use of deep learning methods as the foundation of the developed technique will allow eliminating the major vulnerability of the most often used security methods – identification of unknown forms of malicious software.

It is also shown that the percentage of new threat semantics decreases as a result of the fact that new instances of malicious software are only modifications of already implemented threat mechanisms to which polymorphism and obfuscation have been applied in order to change their signatures. Such a trend is positive, as it allows to significantly increase the security of information systems by preventing the execution of a considerable amount of malicious software in case of the specific approach development which will allow detecting and preventing threats resistant to such modifications.

References

[1] PurpleSec LLC, “The Ultimate List of Cyber Security Statistics for 2022,” *PurpleSec*, 2022. https://purplesec.us/resources/cyber-security-statistics/

[2] StatCounter, “Operating System Market Share Worldwide - September 2022,” *StatCounter Global Stats*, 2022. https://gs.statcounter.com/os-market-share

[3] “Internet Security Report - Q1 2022 | WatchGuard Technologies,” *www.watchguard.com*, Jun. 27, 2022. https://www.watchguard.com/wgrd-resource-center/security-report-q1-2022

[4] “IT threat evolution in Q2 2022. Non-mobile statistics,” *securelist.com*, Aug. 15, 2022. https://securelist.com/it-threat-evolution-in-q2-2022-non-mobile-statistics/107133

[5] “The Five Most Popular Operating Systems for the Internet of Things,” *Open Source for U*, Oct. 31, 2022. https://opensourceforu.com/2019/10/

[6] “IoT Attacks Escalating with a 217.5% Increase in Volume,” *BleepingComputer*, 2022. https://www.bleepingcomputer.com/%20news/security/iot-attacks-escalating-with-a-2175-percent-increase-in-volume

[7] “Margaret Rouse Malware (malicious software),” *SearchSecurity*, 2022. https://searchsecurity.techtarget.com/definition/malware

[8] “2022 Threat Review Report,” *Malwarebytes*, 2022. https://www.malwarebytes.com/resources/

[9] Joxean Koret and E. Bachaalany, *The antivirus hacker’s handbook*. Indianapolis, In: Wiley, 2015, p. 384.

[11] M. Sikorski and A. Honig, *Practical malware analysis: the hands-on guide to dissecting malicious software*. San Francisco: No Starch Press, 2012, p. 8002.

[12] C. C. Elisan and M. Hypponen, *Malware, rootkits & botnets: a beginner's guide*. New York: McGraw-Hill, 2013.

[13] X. Zhang, K. Wu, Z. Chen, and C. Zhang, "MalCaps: A Capsule Network Based Model for the Malware Classification," *Processes*, vol. 9, no. 6, pp. 929–929, May 2021, doi: https://doi.org/10.3390/pr9060929.

[14] "What is a Backdoor Virus? - Definition, Removal & Example," *Study.com*, 2022. https://study.com/academy/lesson/what-is-a-backdoor-virus-definition-removal-example.html

[15] Z. Zuo, Q. Zhu, and M. Zhou, "On the time complexity of computer viruses," *IEEE Transactions on Information Theory*, vol. 51, no. 8, pp. 2962–2966, doi: https://doi.org/10.1109/TIT.2005.851780.

[16] F. Thomas, *Adware: The Only Book You'll Ever Need.* Thomas, 2015, p. 69.

[17] "Total amount of malware and PUA," *AV-ATLAS - Malware & PUA*, 2022. https://portal.av-atlas.org/malware

[18] A. Souri and R. Hosseini, "A state-of-the-art survey of malware detection approaches using data mining techniques," *Humancentric Computing and Information Sciences*, vol. 8, no. 1, p. 3, 2018, doi: https://doi.org/10.1186/s136730180125x.

[19] C. Raghuraman, S. Suresh, S. Shivshankar, and R. Chapaneri, "Static and Dynamic Malware Analysis Using Machine Learning," in *First International Conference on Sustainable Technologies for Computational Intelligence*, Luhach, Ashish Kumar, J. A. Kosa, Poonia, Ramesh Chandra, X. Gao, and D. Singh, Eds., Singapore: Springer Singapore, 2020, pp. 793–806.

[16] C. H. Malin, E. Casey, and J. M. Aquilina, *Malware Forensics*. Syngress, 2008, p. 132.

[20] E. Eilam, *Reversing: secrets of reverse engineering*. Hoboken, N.J.: Wiley, 2013, p. 624.

[21] M. Egele, T. Scholte, E. Kirda, and C. Kruegel, "A survey on automated dynamic malware-analysis techniques and tools," *ACM Comput. Surv.*, vol. 44, Art. no. 2, 2008, doi: https://doi.org/10.1145/2089125.2089126.

[22] Ö. Aslan and R. Samet, "Investigation of Possibilities to Detect Malware Using Existing Tools," in *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, pp. 1277–1284. doi: https://doi.org/10.1109/AICCSA.2017.24.

[23] K. Alzarooni, "Malware variant detection," 2012.

[24] J. Su, D. V. Vasconcellos, S. Prasad, D. Sgandurra, Y. Feng, and K. Sakurai, "Lightweight Classification of IoT Malware Based on Image Recognition," *IEEE Xplore*, Jul. 01, 2018. https://ieeexplore.ieee.org/abstract/document/8377943?casa_token=mYpEAbB5CZMAAAAA:BvrUJmRS2IhmorJDrMjU5J0Qrqz3sfqiqR7IrEu7BsZtm8OVgzgzTSMj4uiOtHs5u-B88ZCp0qbqdA (accessed Jul. 21, 2020).

[25] A. Azmoodeh, A. Dehghantanha, M. Conti, and K. R. Choo, "Detecting cryptoransomware in IoT networks based on energy consumption footprint," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 4, pp. 1141–1152, 2018, doi: https://doi.org/10.1007/s1265201705585.

UDC 004.855.5 DOI: https://doi.org/10.20535/2708-4930.3.2022.265200

OVERVIEW OF OCR TOOLS FOR THE TASK OF RECOGNIZING TABLES AND GRAPHS IN DOCUMENTS

O. Yaroshenko

This study describes OCR tools for recognizing tables and graphs. There is a great demand for solutions that can effectively automate the processing of an extensive array of documents.

Existing OCR solutions can efficiently recognize text, but recognizing graphical elements, such as charts and tables, is still in the making. Solutions that can increase the accuracy of visual data recognition can be valuable for technical document processing, such as scientific, financial, and analytical documents.

Key words: *OCR, PDF files, FastText, detection, recognition, deep learning, technical documents.*

Introduction

In the modern world, every day, a huge number of different documents are translated from paper to electronic form: printed texts, payment orders, customs or tax declarations, ballots, various questionnaires, and many others. Thousands of different electronic document management systems are actively used in almost all spheres of activity.

Thanks to general computerization and the spread of electronic document circulation in various areas of human activity, huge archives of textual and visual information have been accumulated. The global Internet is a continuously expanding electronic archive.

The analysis of modern information systems made it possible to draw a conclusion about the limited possibilities of semantic analysis and image search. Semantic analysis of images means automatically obtaining their semantic descriptions (annotations) and searching in the space of these descriptions (search by content) [4].

Among the search types implemented by information systems, image search by keywords is the closest to meaningful search but has one significant drawback – keywords for images are created by an expert. In all systems of electronic document circulation and systems of entering printed texts, one of the key stages is the recognition of text symbols – the translation of information from graphic form – the result of scanning – into text form. In most cases, the raw document data has noise in it, i.e., unwanted features that make the image hard to perceive. Although these images can be used directly for feature extraction, the accuracy of the algorithm would suffer greatly. This is why image processing is applied to the image to get better accuracy.

Despite the long history of the development of recognition algorithms and the existence of a large number of algorithms, clearly printed texts are recognized well. The problem of recognition in more complex cases is far from being solved [8].

There is a question of further increasing the accuracy of recognizing documents of poor quality; in particular, existing algorithms provide a relatively low accuracy of recognizing texts from graphic images obtained by scanning with small resolutions [1].

It is worth noting the class of problems in which graphic the image cannot be improved by increasing the scanning resolution or changing the scanning parameters. For this, papers (receipts, business cards, reports, internal decrees) are usually processed by a scanner, and OCR software creates searchable PDF files for the required text fragment.

Text recognition systems or OCR systems (Optical Character Recognition) are designed to automatically enter documents into a computer. It can be a page of a book, a magazine, a dictionary, or some kind of document – anything that has already been printed and needs to be converted back to electronic form [3].

Thus, the development of new high-precision text recognition algorithms, as well as the improvement of existing ones, is a potentially useful task.

OCR systems development

The history of the most massive demand for OCR systems began with the "competition" between CuneiForm and FineReader systems of the same version 1.3. According to many independent specialists, CuneiForm was more robust regarding the sum of indicators than [3].

The backbone of the development team of this program was based in the USA. However, unfortunately, even before the release of the CuneiForm 2.0 version, this team practically ceased to exist. Moreover, BIT kept its team of programmers [9]. OCR is used for two main tasks: document archiving and document editing. For this, papers (receipts, business cards, reports, internal decrees) are usually processed by a scanner, and OCR software creates searchable PDF files for the required text fragment [5].

Such programs usually convert a printed table into an Excel file or a paper document into an electronic one that can be edited and used later on a PC. Powerful OCR software can also convert printed text into HTML files. They can immediately be placed on the site for public access.

These tasks include previously created electronic archives of documents in the form of bitmap images, electronic libraries, and text messages. OCR is used for two main tasks: document archiving and editing [6]. When choosing an OCR program, one needs to decide whether it wants it to run automatically, interactively, or in combination with others. With the offline operation, the utility starts working immediately after scanning the document. A few seconds after processing the paper medium, the program produces the final result [2].

Of course, editors built into recognition systems cannot compete with Microsoft Word or Lotus Word Pro. OCR editors – programs are designed in such a way as to simplify the process of eliminating recognition errors and errors as much as possible: the system allows one to observe the "original" graphic image of the document during the editing process [2].

Almost all recognition programs have a built-in spell check system, even at the recognition stage. In the editor, "doubtful" symbols and words not in the dictionary are highlighted in a unique color.

When the document is edited, it can be saved as a file (TXT, RTF). The RTF format (Rich Text Format) is understandable by most word processors (Microsoft Word, Lotus Word Pro, Word Perfect, Lexikon). It allows one to specify information about the design (fonts, paragraphs, illustrations, columns, trimming, tables) [3].

The finished document can be transferred to the editor using the Drag&Drop mechanism or via the Clipboard. If the document contains tables, they can be written as Word tables or directly transferred to Excel spreadsheets.

OCR systems recognize text and various elements (pictures, tables) from an electronic image. The image is usually obtained by scanning a document and, less often, by photographing it. The algorithm of the OCR program processes the received image, areas of text, images, and tables are highlighted, and garbage is separated from the necessary data (Fig 1).

At the next stage, each character is compared with a unique dictionary of characters; if a match is found, this character is considered recognized (Fig. 2). As a result, one gets a set of recognized characters, that is, the desired text. Modern OCR systems are pretty complex software solutions [7].

After all, the text can be littered, distorted, or polluted, and the program must take this into account and handle such situations properly. In addition, modern OCR systems also make it possible to obtain a copy of a printed document in electronic form, preserving formatting, styles, text sizes, and fonts.

Description of the OCR procedure

1. Image pre-processing.
2. Recognition of objects of higher levels.
3. Character recognition
4. Hypothesis structuring. Vocabulary check.
5. Synthesis of an electronic document.

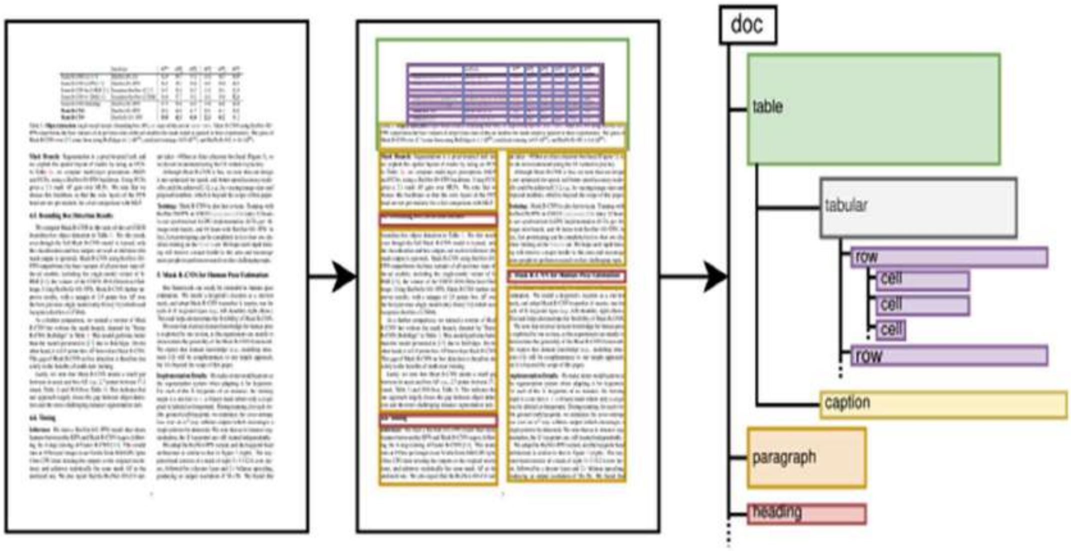


Fig. 1. Document structure recognition [18]

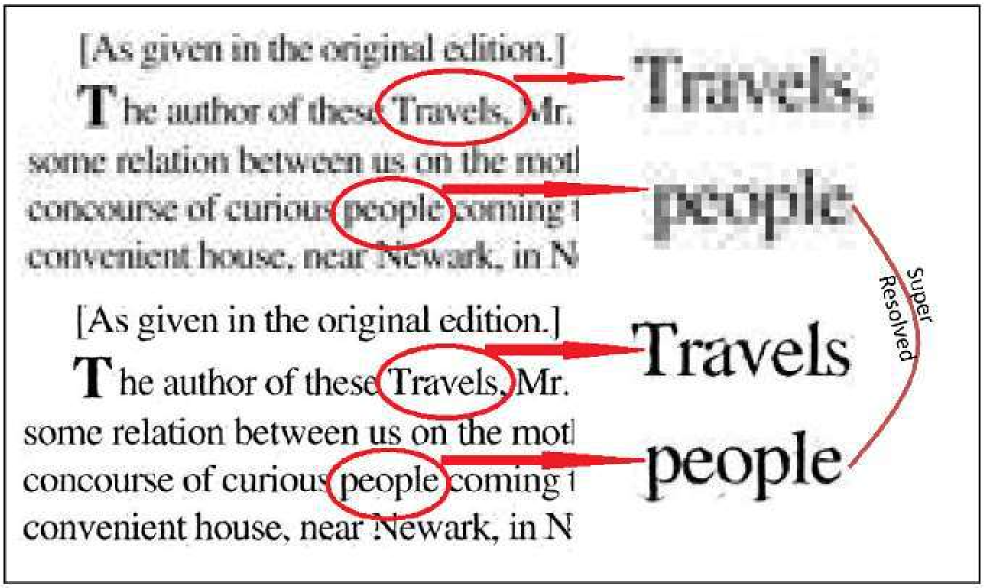

[As given in the original edition.]
The author of these Travels, Mr.
some relation between us on the mot
concourse of curious people coming t
convenient house, near Newark, in N
Travels.
people
Super Resolved
[As given in the original edition.]
The author of these Travels, Mr.
some relation between us on the mot
concourse of curious people coming t
convenient house, near Newark, in N
Travels
people

Fig. 2. Character identification and recognition example [13]

Most OCR Optical Character Recognition programs work with a bitmap image received through a fax modem, scanner, digital camera, or another device. The first step in OCR is to break up the page into blocks of text based on the particularities of right and left alignment and the presence of multiple columns. The recognized block is then split into lines [7].

As a result, there is a problem determining the line to which this or that image fragment belongs. For example, for the letters j, with a slight slope, it is already difficult to determine which line the upper (separate) part of the character belongs to (in some cases, it can be mistaken for a comma or a period). The lines are then broken up into contiguous regions of the image, which generally correspond to individual letters; the recognition algorithm makes assumptions about the correspondence of these regions to characters; and then a selection of each character is made, as a result of which the page is restored in characters of text, and, as a rule, in the appropriate format.

OCR systems can achieve the best recognition accuracy of over 99.9% for pure images composed of regular fonts [4].

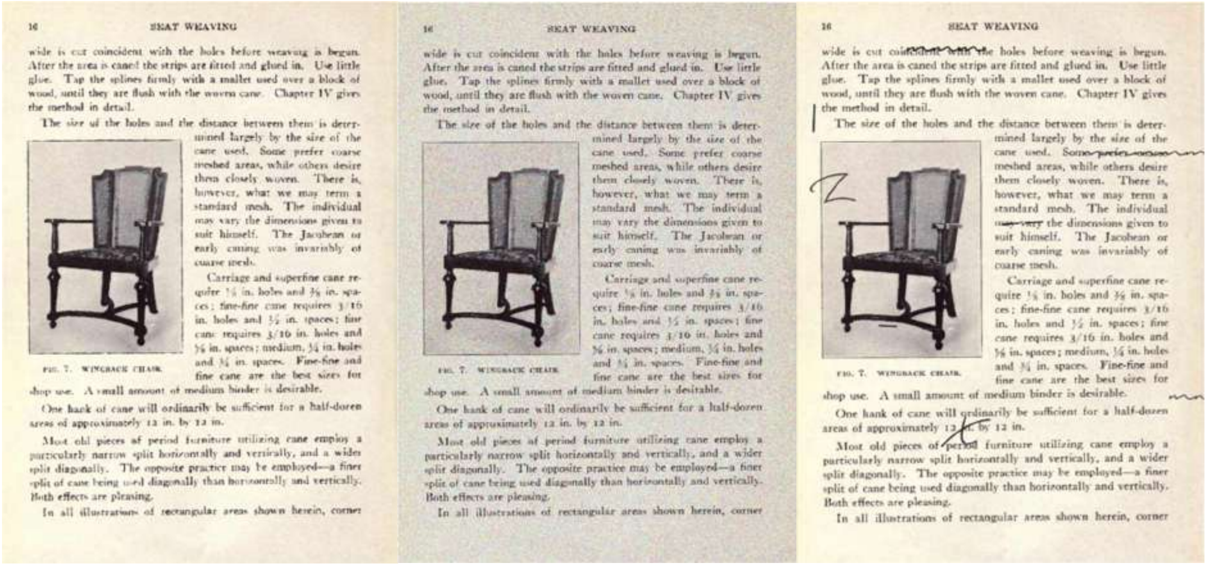
At first glance, this recognition accuracy seems ideal. However, the error rate is still depressing because if there are approximately 1500 characters per page, then even with a recognition success rate of 99.9%, there are one or two errors per page. In such cases, the dictionary check method comes to the rescue.

If a word is not in the system's dictionary, it tries to find a similar one according to special rules. However, it still does not allow to correct 100% of errors, which requires human control of the results.

The modern state of OCR processing for technical documents

Heavy use of PDF files has promoted research in analyzing the file layout for text extraction purposes. One of the PDF document's difficulties is that smartphone users extensively scan the documents in PDF format using the phone camera. Optical Character Recognition (OCR) techniques must be employed to get these images into text format [11]. OCR is a technology still in the making, and available software provides varying levels of accuracy. The best results are usually obtained with a tailored solution involving corpus-specific pre-processing, model training, or postprocessing, but such procedures can be labor-intensive. Pre-trained, general OCR processors have a much higher potential for wide adoption in the scholarly community; hence, their out-of-the-box performance is of scientific interest. Modern OCR framework comparison research indicated that certain types of "integrated" noise, such as blur and salt and pepper, generate more errors than "superimposed" noise, such as watermarks, scribbles, and even ink stains (Fig 3).

Furthermore, it suggests that the "OCR language gap" persists. Calls for special efforts to improve the quality of document images before passing them to the OCR engine. [12] One compelling option is to super-resolve these low-resolution document images before passing them to the OCR engine. Experiments show an improvement of up to 21% in accuracy OCR on test images scanned at low resolution. One immediate application of this can be enhancing the recognition of historical documents scanned at low resolutions [13].



16 SEAT WEAVING

wide is cut coincident with the holes before weaving is begun. After the area is caned the strips are fitted and glued in. Use little glue. Tap the splines firmly with a mallet used over a block of wood, until they are flush with the woven cane. Chapter IV gives the method in detail.

The size of the holes and the distance between them is determined largely by the size of the cane used. Some prefer coarse meshed areas, while others desire them closely woven. There is, however, what we may term a standard mesh. The individual may vary the dimensions given to suit himself. The Jacobean or early caning was invariably of coarse mesh.

FIG. 7. WINGBACK CHAIR

Carriage and superfine cane require ⅛ in. holes and ⅜ in. spaces; fine-fine cane requires 3/16 in. holes and ½ in. spaces; fine cane requires 3/16 in. holes and ⅝ in. spaces; medium, ¼ in. holes and ¾ in. spaces. Fine-fine and fine cane are the best sizes for shop use. A small amount of medium binder is desirable.

One hank of cane will ordinarily be sufficient for a half-dozen areas of approximately 12 in. by 12 in.

Most old pieces of period furniture utilizing cane employ a particularly narrow split horizontally and vertically, and a wider split diagonally. The opposite practice may be employed—a finer split of cane being used diagonally than horizontally and vertically. Both effects are pleasing.

In all illustrations of rectangular areas shown herein, corner

Fig. 3. Document noise examples

Scientific papers and other technical documents are composed of natural language text and other modalities, like block diagrams, mathematical formulas, tables, graphics, and pictures. Automatic Technical Documents Processing and Understanding (TDPU) has received more attention in the last two decades due to its profound applicability. TDPU represents the continuation of the progress made in the fields of OCR, Natural Language Understanding, Pattern Recognition, and Image Understanding. [14]

Research activities in document image analysis can be mainly classified into two categories, text processing, and non-text processing, e.g., figures, graphics, and diagrams [19]. Although the introduction of optical character recognition technologies mostly solved the task of converting human-readable characters from images into machine-readable characters, the task of extracting table semantics has been less focused on over the years [16]. Also, chart recognition techniques for document images are still an unsolved problem due to the great subjectiveness and variety of chart styles [19].

The recognition of tables consists of two main tasks, namely table detection and table structure recognition (Fig 4). There are works that recognize table structures from text or other syntactic tokens rather than directly from document renderings. One draws upon deep neural networks to identify table structures for rendered inputs. The proposed architecture combined the benefits of convolutional neural networks for visual feature extraction and graph networks for dealing with the problem structure. They empirically demonstrated that their method outperforms the baseline by a significant margin. However, they aim at a different purpose: parsing table structures but not complete document hierarchies. As such, the authors do not attempt to identify text elements or nested figures. [17].

Research regarding mathematical formula detection identified that the key difference between formula detection in typeset documents and object detection in natural scenes is that typeset documents avoid the occlusion of content by design. This constraint may help design a better algorithm for non-maximal suppression, as the original non-maximal suppression algorithm is designed to handle overlapping objects. They believe improved pooling will reduce the number of over-merged and split detections, improving precision and recall. This approach can detect not only formulas but also other types of structures in technical documents [10].

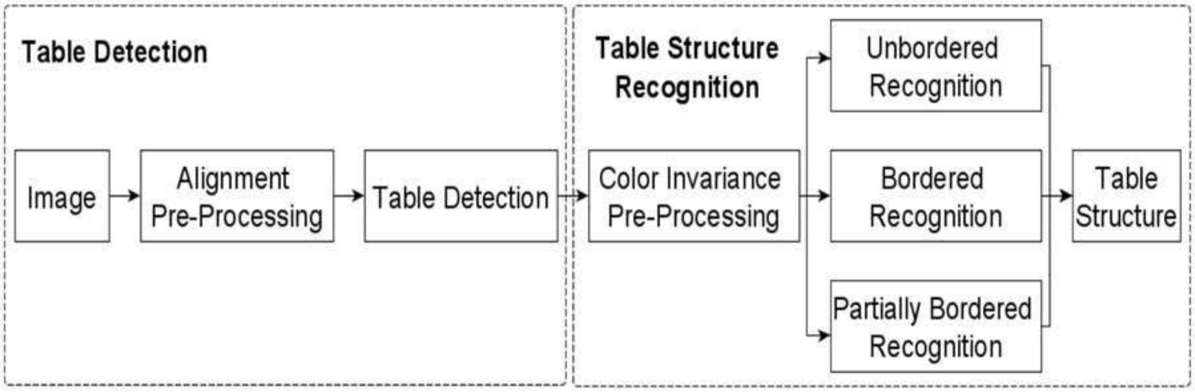
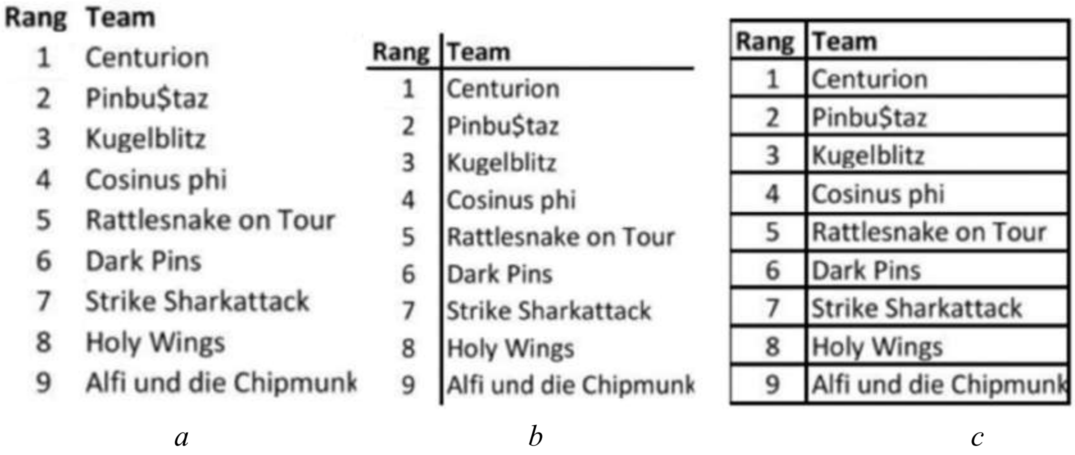


Fig. 4. The two-stage process of TD and TSR in Multi-Type-TD-TSR. [16]

Prasad et al. [15] developed a model for table structure detection based on CNN architecture which was originally trained for objects in natural scene images and was also very effective for detecting tables. Moreover, iterative transfer learning and image augmentation techniques can be used to learn efficiently from a small amount of data. The proposed model recognized structures within tables by predicting table cell masks while using the line information. It was stated that improving the post-processing modules can further enhance the accuracy [15].

For this purpose, Fisher et al. [16] distinguished three types of tables (Fig 5), depending on whether they are borderless or not. Because of the unavailability of large labeled datasets for table structure recognition, they decided to use two conventional algorithms: The first one that can handle tables without borders and the second one that can handle tables with borders. Further, they combined both algorithms into a third conventional table structure recognition algorithm that can handle all three types of tables. This algorithm achieves the highest F1 score among the systems compared in their research for an IoU threshold of 0.6 and 0.7 but does not detect sharp borders, so the F1-score decreases rapidly for higher thresholds of 0.8 and 0.9 [16].



Rang	Team
1	Centurion
2	Pinbu$taz
3	Kugelblitz
4	Cosinus phi
5	Rattlesnake on Tour
6	Dark Pins
7	Strike Sharkattack
8	Holy Wings
9	Alfi und die Chipmunk

a

Rang	Team
1	Centurion
2	Pinbu$taz
3	Kugelblitz
4	Cosinus phi
5	Rattlesnake on Tour
6	Dark Pins
7	Strike Sharkattack
8	Holy Wings
9	Alfi und die Chipmunk

b

Rang	Team
1	Centurion
2	Pinbu$taz
3	Kugelblitz
4	Cosinus phi
5	Rattlesnake on Tour
6	Dark Pins
7	Strike Sharkattack
8	Holy Wings
9	Alfi und die Chipmunk

c

Fig. 5. Types of tables based on how they utilize borders: *a* – tables without borders, *b* – tables with partial borders, *c* – tables with borders [16]

Rausch et al. [18] presented a solution that takes rendered document images as input, performs segmentation into bounding boxes, and then outputs the hierarchical structure of the entire document. Their solution identified table and tabular elements with high precision, but other elements were recognized with significantly lower accuracy. They emphasize again that both suitable baselines and datasets for this task are hitherto lacking [18]. Qasim et al. [17] also identified the lack of large-scale datasets as a significant hindrance to deep learning research for structure analysis. They presented a new large-scale synthetic dataset for the problem of table recognition [17]. Ayinala and Grandhi (2021) stated that text processing is an essential task as we have more digital content available on the Internet today. The most challenging task nowadays is locating and analyzing textual information [11].

Another big problem with the correct processing of technical documents is chart recognition. They are widely used to represent numeric and qualitative data in different formats. Although existing OCR tools can recognize the text content of chart segments, the primary data represented by the chart, which is usually shown visually by lines, bars, and circle segments, is not recognized well by those tools. The main issue is that there are plenty of different chart types and styles for each particular chart type, and most research is focused on a limited set of charts representation [19].

For the task of extracting data from chart images, the detection process is a preliminary step. It helps to locate and extract the data chart only and classify chart type, improving data recognition performance. For such tasks, Convolutional Neural Networks (CNN) are commonly used. CNN-based methods show outstanding results in various object detection domains. There is a lack of works in the literature linking real-world photos with the task of labeling charts before labeling. There are many issues to solve, such as locating charts in images and removing camera distortions [20].

It can be said that more advanced solutions for chart recognition are a necessary addition to existing OCR systems [19].

Conclusion

OCR systems recognize text and various elements (pictures, tables) from an electronic image. The image is usually obtained by scanning a document and, less often, by photographing it. The algorithm of the OCR program processes the received image, areas of text, images, and tables are highlighted, and garbage is separated from the necessary data. At the next stage, each character is compared with a unique dictionary of characters; if a match is found, this character is considered recognized. As a result, one can get a set of recognized characters, that is, the desired text.

As described above, technical document processing is a demanding and underdeveloped area of deep learning. There have been many types of research in this area in recent years, but the main focus is on the text and common pattern recognitions inside documents. On the other hand, technical documents have a lot of specific structures (mainly charts and tables) inside and require a high recognition accuracy to be considered.

Directions for future research

There are several gaps in the technical document processing research that follow from the findings in this article that would benefit from further research, including extending and further testing statements developed here:

1. In-depth exploration of how OCR algorithms can be re-evaluated and modified to perceive scanned PDFs with technical documentation better. That may include new approaches to removing noise from scanned images and solutions for document content structure recognition.

2. Gathering new datasets of scanned PDFs that include specific elements, such as tables and charts of different types, to enhance further models that work with processing such elements.

3. Improing unique structure recognition and processing methods by comparing existing deep neural networks with different architectures for such tasks. Based on the results, it would be beneficial to build data pipelines that will combine different methods to improve the final solution's accuracy.

References

[1] A. L. Gorelik and V. A. Skripkin, *Methods of Recognition (Studies in Cybernetics*. High School, 1984, p. 219.

[2] G. Ya. Voloshyn and A. A. Ilyin, *Pattern recognition methods. Book 2*.

[3] A. N. Pisarevskiy, A. F. Chernyavskiy, and G. K. Afanas'ev., "Systems of technical vision (fundamental principles, hardware and software)," *Mechanical engineering. Leningrad department*, p. 423, 1988.

[4] V. P. Babak, V. S. Khandetsky, and E. Schryufer, *Obrobka signals: Handyman*. Kyiv: Libid, 1996.

[5] G. P. Vyatkina, "Engineering drawing," *Mechanical engineering*, p. 368, 1985.

[6] D. Blatner, G. Fleishman, and S. Rot, *Scanning and rasterization of images*. EKOM Publishing House, 1999, p. 400.

[7] D. Forsyth and J. Ponce, *Computer vision: a modern approach*. Williams Publishing Center, 2004, p. 928.

[8] E. V. Mikheeva, *Information technology in professional activities*. Academy, 2007, p. 384.

[9] E. I. Grebenyuk and N. A. Grebenyuk, *Technical means of informatization: Textbook for environments. Prof. education*. Publishing Center "Academy," 2005, p. 272.

[10] P. Mali, P. Kukkadapu, M. Mahdavi, and R. Zanibbi, "ScanSSD: Scanning Single Shot Detector for Mathematical Formulas in PDF Document Images," *arXiv.org*, Mar. 17, 2020. https://arxiv.org/abs/2003.08005

[11] H. K. Ayinala and S. Grandhi, "Text classification from PDF documents," *In International Research Journal of Modernization in Engineering Technology and Science*, vol. 3, pp. 58–63, 2021.

[12] T. Hegghammer, "OCR with Tesseract, Amazon Textract, and Google Document AI: a benchmarking experiment," *Journal of Computational Social Science*, vol. 5, no. 1, pp. 861–882, 2022, doi: https://doi.org/10.1007/s42001021001491.

[13] A. Lat and C. V. Jawahar, "Enhancing OCR Accuracy with Super Resolution," in *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 3162–3167. doi: https://doi.org/10.1109/ICPR.2018.8545609.

[14] E. E. Kostalia, M. Petrakis, and N. Bourbakis, "Evaluating Methods for the Parsing and Understanding of Mathematical Formulas in Technical Documents," in *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 407–412. doi: https://doi.org/10.1109/ICTAI50040.2020.00070.

[15] D. Prasad, A. Gadpal, K. Kapadni, M. Visave, and K. Sultanpure, "CascadeTabNet: An approach for end to end table detection and structure recognition from imagebased documents," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2439–2447. doi: https://doi.org/10.1109/CVPRW50498.2020.00294.

[16] P. Fischer, A. Smajic, G. Abrami, and A. Mehler, “MultiTypeTDTSR – Extracting Tables from Document Images Using a Multistage Pipeline for Table Detection and Table Structure Recognition: From OCR to Structured Table Representations,” in *KI 2021: Advances in Artificial Intelligence*, S. Edelkamp, R. Möller, and E. Rueckert, Eds., Cham: Springer International Publishing, 2021, pp. 95–108.

[17] S. R. Qasim, H. Mahmood, and F. Shafait, “Rethinking Table Recognition using Graph Neural Networks,” in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 142–147. doi: https://doi.org/10.1109/ICDAR.2019.00031.

[18] J. Rausch, O. Martinez, F. Bissig, C. Zhang, and S. Feuerriegel, “DocParser: Hierarchical Document Structure Parsing from Renderings,” in *The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)*, 2021, pp. 4328–4338.

[19] Y. Liu, X. Lu, Y. Qin, Z. Tang, and J. Xu, “Review of chart recognition in document images,” *Proceedings of SPIE*, vol. 8654, Feb. 2013, doi: https://doi.org/10.1117/12.2008467.

[20] T. Araújo, P. Chagas, J. Alves, C. Santos, S. Santos, and Serique Meiguins, Bianchi, “A RealWorld Approach on the Problem of Chart Recognition Using Classification, Detection and Perspective Correction,” *Sensors*, vol. 20, no. 16, 2020, doi: https://doi.org/10.3390/s20164370.

ABSTRACTS

UDC 004.383
DOI: https://doi.org/10.20535/2708-4930.3.2022.267302

DESIGN OF DATA BUFFERS IN FIELD PROGRAMMABLR GATE ARRAYS

(p. 4 – 16)

Anatoliy Sergiyenko
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine.
ORCID: http://orcid.org/0000-0001-5965-1789

Pavlo Serhiienko
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine.
ORCID: http://orcid.org/0000-0003-3030-0074

Ivan Mozghovyi
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine.
ORCID: http://orcid.org/0000-0001-5469-486X

Anastasiia Molchanova
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine.
ORCID: http://orcid.org/0000-0001-7328-7151

The need to intensify the extraction process using the influence of chemical reagents on beet chips was substantiated. The analysis of application of natural sorbents in food production technologies was carried out. The physical and chemical properties of zeolite were explored. The indicators that make it possible to apply natural zeolite for additional treatment of water and juices in sugar production were shown. The effectiveness of the use of natural zeolite for feed water treat-ment with the view to enhancing the technological quality of diffusive juice was determined. Experimental research revealed that feed water treatment with zeolite decreases the content of total iron, ammonium, and permanganate oxidation indicator. It was proved that microbial seeding of feed water and diffusive juice decreases in case of treatment with zeolite. It was established experimentally that the purification of diffu-sion juice occurs during zeolite application for feed water treatment. We determined the effectiveness of removal of macromolecular compounds, including dextran, from diffusive juice obtained during processing sugar beets of various technological quality with natural zeolite. It was shown that at the zeolite consumption of 0.1...0.4 % to the weight of beets, the content of high-molecular compounds and pectic substances in diffusive juice decreases by 30–40 %, and the content of dextran – by 20–40 %, respectively. During the zeolite treatment, an enhancement of the quality of purified juice and improvement of filtration and saturation proper-ties of defeco-saturated precipitate are observed. Thus, the average rate of sedimentation of the precipitate of juice of I carbonation S5 m, when using zeolite for feed water preparation increases by 10–50 % for the beet different technological quality. In the course of research, we designed the technique of zeolite application, which ensures a decrease in coloration, an increase in the purity of the cleared juice, enhancement of filtration and sedimentation properties of the precipitate of juice of I carbon-ation. High effectiveness of the proposed method is pronounced in processing raw materials of lowered quality. Thus, there are some grounds to claim the effectiveness of zeolite application to enhance the quality of diffusion juice and products in sugar production.

Keywords**:** diffusion juice, dextran, sucrose extraction, purification of diffusion juice, zeolite.

UDC 004.056.5
DOI: https://doi.org/10.20535/2708-4930.3.2022.265480

ORGANIZATION OF FAST EXPONENTIATION ON GALOIS FIELDS FOR CRYPTOGRAPHIC DATA PROTECTION SYSTEMS

(p. 17 – 25)

Al-Mrayt Ghassan Abdel Jalil Halil
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine
ORCID: http://orcid.org/0000-0002-4382-5309

Oleksandr Markovskyi
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine
ORCID: http://orcid.org/0000-0003-3483-4233

Alona Stupak
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine.
ORCID: http://orcid.org/0000-0002-3491-7365

The object of the research described in the article is the process of calculating the exponent on finite Galois fields when implementing cryptographic mechanisms for protecting information with a public key.

The purpose of these studies is to speed up the exponentiation operation on Galois fields, which is basic for the implementation of a wide range of cryptographic data protection protocols through the use of precomputations that depend only on the forming Galois polynomial field.

To achieve the goal, the feature of performing exponentiation on Galois fields in public key cryptography is used – the constancy of the forming Galois field polynomial, which is part of the public key. This allows you to select calculations that depend only on the generating polynomial and perform them only once, saving the results in the precalculation tables. The use of precomputations allows not only to reduce the computational complexity of the exponentiation operation on Galois fields, but also to effectively use it to speed up the combination of the processing of several bits.

The article proposes the organization of accelerated execution of the basic operation of a wide range of cryptographic algorithms with a public key – exponentiation on finite Galois fields GF(2^n). Acceleration of the computational implementation of this operation is achieved by organizing the processing of several bits of the code at once during squaring on Galois fields. This organization is based on the use of polynomial squared properties, Montgomery group reduction, and extensive use of previous calculations. Procedures for performing basic operations of exponentiation on Galois fields are developed in detail, the work of which is illustrated by numerical examples. It has been proved that the proposed organization can increase the computational speed of this operation by 2.4 times, which is significant for cryptographic applications.

Keywords: multiplication operation on Galois fields, cryptographic algorithms based on Galois Fields algebra, Galois Fields exponentiation, Montgomery reduction.

UDC 004. 004.4 (043.2)
DOI: https://doi.org/10.20535/2708-4930.3.2022.265418

ORGANIZATION OF PARALLEL EXECUTION OF MODULAR MULTIPLICATION TO SPEED UP THE COMPUTATIONAL IMPLEMENTATION OF PUBLIC-KEY CRYPTOGRAPHY

(p. 26 – 32)

Igor Boiarshyn
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine
ORCID: http://orcid.org/0000-0002-5318-8234

Oleksandr Markovskyi
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine
ORCID: http://orcid.org/0000-0003-3483-4233

Bohdana Ostrovska
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine
ORCID: http://orcid.org/0000-0001-7967-4582

The object of research to which the article is devoted are the processes of calculating multiplicative operations of modular arithmetic, which are performed on numbers, the length of which is orders of magnitude greater than the bit capacity of processors.

The target of the research is to speed up the execution of the modular multiplication operation on numbers, which is important for cryptographic tasks, the bit count of which significantly exceeds the bit count of the processor, due to the organization of parallel calculation of fragments of the modular product on multi-core computers.

As the main way to achieve the goal, in the research presented in the article, parallelization at the level of processing bits of the multiplier and the application of Montgomery group reduction using recalculations that depend only on the module, which for cryptographic applications is part of the public key, which allows it to be considered constant, were used.

The article theoretically substantiates, develops and investigates the method of parallel execution of the basic operation of cryptography with a public key – modular multiplication of large numbers. It is based on a special organization of dividing the components of modular multiplication by independent computational processes in order to ensure the possibility of effective group reduction of the product. The proposed organization ensures high independence of partial computing processes, which simplifies the organization of interaction between them. To implement the Montgomery group reduction, the results of recalculations are used, which depend only on the module and, accordingly, are performed only once. The presentation is illustrated by numerical examples. It is theoretically and experimentally proven that the proposed approach to the parallelization of the computational process of modular multiplication using Montgomery group reduction when using s processor cores allows to speed up this important for cryptographic applications operation by 0.57·s.

Key words: modular multiplication, Montgomary modular reductions, open key cryptography, parallel computation, multiplicative operations of modular arithmetic.

UDC 004
DOI: https://doi.org/10.20535/2708-4930.3.2022.267665

SIMULATION OF FLUID MOTION IN COMPLEX CLOSED SURFACES USING A LATTICE BOLTZMANN MODEL

(p. 33 – 41)

Valentyn Kuzmych
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine
ORCID: http://orcid.org/0000-0002-6077-3609

Mykhailo Novotarskyi
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine
ORCID: http://orcid.org/0000-0002-5653-8518

Restorative operations on the human digestive tract can cause negative consequences. These effects were manifested in the appearance of undesirable deformations, so-called "blind bags", which arose as a result of the formation of high pressure zones after the geometry of the digestive tract cavity was changed during reconstructive surgery. For this reason, the development of a mathematical model of the movement of liquids in closed surfaces has become necessary in recent years.

There are many approaches to solving such problems. Most of the traditional approaches require considerable time and computing resources for their implementation. Thus, when using analytical methods, there are possible solutions only under certain conditions. Therefore, there is a need to use effective numerical methods, one of which is the lattice Boltzmann model. The purpose of this work is to study hydrodynamic processes in closed surfaces using the lattice Boltzmann model.

There is a significant number of publications devoted to modeling the movement of liquids in closed surfaces. But the analysis of the publications showed that the use of the Boltzmann lattice model in such problems has not been studied in detail.

Formulated statement of the problem in the form of a numerical solution of the Boltzmann equation. It is proposed to solve such a problem by using the Boltzmann lattice model.

The results of theoretical studies and experiments showed the practical feasibility of using the proposed approach to modeling the movement of liquids in closed surfaces. It is noted that the proposed approach has prospects for further research and development.

***Key words**: hydrodynamics, lattice Boltzmann model.*

UDC 004.052.42

DOI: https://doi.org/10.20535/2708-4930.3.2022.269112

ZERO-KNOWLEDGE IDENTIFICATION OF REMOTE USERS BY UTILIZATION OF PSEUDORANDOM SEQUENCES

(p. 42 – 48)

Ihor Daiko
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine
ORCID: http://orcid.org/0000-0002-5316-7080

Viktor Selivanov
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine
ORCID: http://orcid.org/0000-0001-8519-6038

Miroslava Chernyshevych
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine
ORCID: http://orcid.org/0000-0002-5033-3028

Oleksandr Markovskyi
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine
ORCID: http://orcid.org/0000-0003-3483-4233

The object of the research described in the article is the process of cryptographically strict identification of participants in remote information interaction, which provides the possibility of protection against session interception by outsiders.

The purpose of the work is to increase the effectiveness of cryptographically strict identification of participants in remote information interaction due to the acceleration of the identity confirmation process, as well as by organizing secondary cycles of contact control to counteract interaction interception.

The goal is achieved through the additional use of secondary identification cycles, which are carried out periodically during the interaction session and allow detecting the fact that the interaction was intercepted by the attacker. A single cryptographic mechanism – generators of pseudo-random binary sequences – is used to implement primary and secondary identification. In addition to implementing cryptographically strict identification, this mechanism can be used for fast stream encryption of data exchanged by participants of interaction.

In the article, the identification scheme based on the concept of "zero knowledge" with the use of irreversible generators of pseudo-random bit sequences is theoretically justified, developed and researched in detail. Session passwords form a chain formed by random sequence values. Secondary identification sessions are provided in the proposed scheme to counteract attacks with the imprinting of one of the remote interaction parties. The main elements of the proposed identification scheme are developed in detail: authorization procedures, primary and secondary identification.

It is theoretically proven that the task of breaking the proposed method of cryptographically strict identification is identical to the prediction of the binary sequence formed by the generator. For standardized cryptographic generators of pseudorandom sequences, the solution of this problem is beyond the technical capabilities for most practical applications. It is theoretically and experimentally proven that the proposed cryptographically strict identification scheme provides 2 – 3 orders of magnitude faster performance compared to known schemes that use irreversible number theory transformations and is an order of magnitude faster than identification schemes based on a chain of hash transformations.

***Key words**: Zero-knowledge identification, chain of passwords, cryptographically strong identification, generators of pseudo-random bit sequences, middle attacks.*

UDC 004.052.42
DOI: https://doi.org/10.20535/2708-4930.3.2022.269132

ORGANIZATION OF PROTECTED FILTERING OF IMAGES IN CLOUDS

(p. 49 – 55)

Alireza Mirataei
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine
ORCID: http://orcid.org/0000-0002-4732-7030

Rusanova Olga
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine
ORCID: http://orcid.org/0000-0003-3511-4438

Tribynska Karolina
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine
ORCID: http://orcid.org/0000-0001-8268-2372

Oleksandr Markovskyi
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine
ORCID:http://orcid.org/0000-0003-3483-4233

The object of research is the processes of homomorphic encryption of images for their protected arithmetic mean filtering in clouds.

The purpose of the work is to increase the efficiency of secure image processing in the clouds, in particular, their arithmetic mean filtering on remote computer systems by increasing the level of security.

The article proposes an approach to using cloud technologies to accelerate the filtering of image streams while ensuring their protection during processing on remote computer systems. Homomorphic encryption of images during their remote filtering is proposed to be carried out by shuffling rows of pixel matrices.

This approach is specified in the form of a method of homomorphic encryption of images to protect against their illegal reconstruction during arithmetic mean filtering on remote computer systems, which is distinguished by the fact that the main element of protection is the shuffling of image pixel matrix rows. The shuffling order can change randomly and serves as a secret key for homomorphic encryption of images. Within the framework of the developed method, procedures for partial arithmetic mean filtering, which is carried out on remote systems, as well as procedures for the final stage of filtering, which is carried out on a terminal platform that performs processing and analysis of a real image, are defined. The developed method of protected filtering based on shuffling the rows of the pixel matrix allows, due to the use of remote computing power, to speed up this operation by 1 – 2 orders of magnitude, which practically coincides with the similar indicator of the fastest-acting variant of image protection based on additive masking.

The main advantage of the developed method is a much higher level of protection against attempts, using statistical analysis, to gain illegal access to images during their processing on remote computer systems not controlled by the user.

The proposed method can be used to speed up the processing and analysis of images by terminal devices of computer systems for remote monitoring of the state of real-world objects and their management.

Key words: Arithmetic mean filtration, images processing, homomorphic encryption, secure clouds computing.

UDC 004.056.5

DOI: https://doi.org/10.20535/2708-4930.3.2022.265479

FAST SECURE CALCULATION OF THE OPEN KEY CRYPTOGRAPHY PROCEDURES FOR IOT IN CLOUDS

(p. 56 – 62)

Alireza Mirataei
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine
ORCID: http://orcid.org/0000-0002-4732-7030

Maria Haidukevych
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine
ORCID: http://orcid.org/0000-0003-2334-2401

Oleksandr Markovskyi
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine
ORCID:http://orcid.org/0000-0003-3483-4233

The object of the research described in the article is the process of protected implementation on remote computer systems of the basic operation of public key cryptography - modular exponentiation.

The aim of the study is to increase the speed of implementation of public key cryptographic data protection mechanisms on terminal microcontrollers of computer control systems in real time by organizing the secure execution of the basic operation of these mechanisms - modular exponentiation on remote computer systems.

To attain these aims, the multiplicative-additive decomposition of the exponent code was used, which allowed to divide the calculation into two parts, the larger of which is performed on remote computer systems using cloud technologies, and the smaller one on the terminal microcontroller. At the same time, it is almost impossible to recover the secret components of the operation based on the data transmitted to the cloud on remote computer systems.

As a result of the conducted research, a method for accelerating the implementation of cryptographic data protection mechanisms on built-in IoT terminal microcontrollers, the basic operation of which is modular exponentiation of large-bit numbers, was theoretically justified and developed. The method is based on the use of remote computer systems to speed up calculations and provides protection against the reconstruction of secret keys of cryptosystems based on data transmitted to the cloud. The main difference of the proposed method is the use of a single mechanism for protecting the secret components of the operation in the form of a multiplicative decomposition of the exponent code.

Theoretically and experimentally, it has been proven that the method allows to speed up the execution of cryptographic data protection protocols in IoT by an average of 50 times while providing a level of security sufficient for most practical applications.

***Key words**: Modular exponentiation, secure cloud computing, IoT security, RSA cryptosystems.*

UDC 004.02, 004.2

DOI: https://doi.org/10.20535/2708-4930.3.2022.265229

METHODS OF EFFECTIVIZATION OF SCALABLE SYSTEMS: REWIEW

(p. 63 – 76)

Oleksandr Honcharenko
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine
ORCID: http://orcid.org/0000-0002-9086-6988

Heorhii Loutskii
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine
ORCID: http://orcid.org/0000-0002-3155-8301

The article discusses the problem of inefficiency of modern systems and horizontal scaling as a method of increasing productivity. The main issues that make up the mentioned problem are highlighted: parallelism constraints, mismatch between the task and the system, the complexities of programming and the question of the balance between cost and performance. A classification for possible solutions was proposed, according to which they were divided into architectural and network, and an overview was carried out. As part of the architectural class, such approaches as quantum computing and the dataflow paradigm were reviewed, the most promising solutions were analyzed.

The comparative analysis shows that by their nature dataflow and quantum computing do not contradict each other, moreover, they complement each other in the context of the problem. Thus, specialized D-Wave quantum computers, in contrast to universal quantum processors, provide large computing power at a relatively modest price, while the dataflow solution, represented mainly by Maxeler processors, is universal and efficient, but inferior to quantum systems in a number of tasks.

At the same time, both types of processors require a certain network for communication, which makes the issue of topology relevant. At the network level, 2 topologies – Fat Tree and Dragonfly – were considered, and their main properties were highlighted. The analysis showed that in the context of the problem Dragonfly is slightly better due to decentralization and smaller diameter, however, both solutions provide good topological characteristics and support for the main modern routing technologies.

In the conclusions, the main aspects of problem formulation and review are indicated, further prospects and possible methods are considered. First of all, a promising idea is the combination of quantum and non-quantum solutions in one system. This approach allows you to significantly speed up certain calculations, while ensuring the universality of the system. However, a more general issue is the mutual integration of solutions as such. The problem of efficiency has many partial solutions, but not all of them are compatible, therefore, the development of complex methods on the basis of already known ones is a key perspective of the subject area.

***Keywords**: effectivization, scalable systems, high performant computing, architecture, topology.*

UDC 004.056.5
DOI: https://doi.org/10.20535/2708-4930.3.2022.266391

MODERN INFORMATION SYSTEMS SECURITY MEANS

(p. 77 – 86)

Iryna Klymenko
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine
ORCID: http://orcid.org/0000-0001-5345-8806

Anna Verner
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine
ORCID: http://orcid.org/0000-0001-8598-363X

High rates of technical progress and the spread of information technologies are a fairly widespread phenomenon today. However, statistical data indicate that, simultaneously with the positive dynamics, there is also an annual exponential growth in the amount of malicious software that affects information systems. Thus, in the second quarter of 2022, security systems detected 55.3 million malicious and potentially unwanted objects, which became a serious threat to information security, taking various forms, including attacks on software, theft of intellectual property, theft of personal data, theft of information, sabotage and extortion of information. That is why technologies for analysis and detection of potential dangers are constantly being improved. However, currently no method is capable of detecting the entire existing spectrum of malicious software, which proves the complexity and necessity of creating effective approaches to detecting malicious software and the presence of an unlimited space of possibilities for the development of new methods in this field.

This article reviews the actual state of information security, classifies and highlights specific attributes of security mechanisms, analyzes various criteria for classifying information system security risks.

In the first chapter, categories and features of types of threats to information security are considered. The second chapter provides a general description of threat analysis methods, compares static, dynamic, and hybrid malware analysis methods and highlights the advantages and disadvantages of each of them.

In the third chapter, the newest means of detecting and countering threats to information systems are considered, and the peculiarities of their implementation are analyzed.

The article provides a thorough review of current research on malware detection methodology

The purpose of this article is to provide a general idea of the current state of information security and existing modern methods of protecting information systems from possible threats.

***Key* words**: *information security, information systems, security means, malware.*

UDC 004.855.5
DOI: https://doi.org/10.20535/2708-4930.3.2022.265200

OVERVIEW OF OCR TOOLS FOR THE TASK OF RECOGNIZING TABLES AND GRAPHS IN DOCUMENTS

(p. 87 – 94)

Oleksandr Yaroshenko
National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, Ukraine
ORCID: http://orcid.org/0000-0003-1871-3810

This study provides an overview of OCR tools for recognizing document tables and graphs. Digitizing paper documents has many advantages for both individuals and businesses. One must use OCR (optical character recognition) software to digitize. Such software scans documents to make the text readable by a computer. One can convert them to formats supported by Microsoft Word or Google Docs. OCR software is becoming more of a necessity than a utility for entertainment. OCR creates searchable, editable text from printed documents, as well as from scanned photos or books and PDF files.

Currently, there is an active trend toward the digitalization of documents. There is a great demand for solutions that can effectively automate the processing of an extensive array of documents with high accuracy. A particular case is the processing of PDF files, such as scanned documents or generated by software editors. OCR solutions aim to increase the efficiency of processing and analysis of digital documents using artificial intelligence. Both government agencies and businesses can use these solutions. The developed systems can be a valuable addition to CRM systems and can be integrated instead of existing document processing modules or used as a separate solution.

Although existing OCR solutions can efficiently recognize text, recognizing graphical elements, such as charts and tables, is still in the making. Solutions that can increase the accuracy of visual data recognition can be valuable for technical document processing, such as scientific, financial, and other analytical documents.

***Key words**: OCR, PDF files, FastText, detection, recognition, deep learning, technical documents.*

АНОТАЦІЇ

УДК 004.383
DOI: https://doi.org/10.20535/2708-4930.3.2022.267302

РОЗРОБКА БУФЕРІВ ДАНИХ НА ПРОГРАМОВАНИХ ЛОГІЧНИХ ІНТЕГРАЛЬНИХ СХЕМАХ

(Стор. 4 – 16)

Анатолій Сергієнко
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0001-5965-1789

Павло Сергієнко
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0003-3030-0074

Іван Мозговий
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0001-5469-486X

Анастасія Молчанова
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0001-7328-7151

Стаття присвячена вирішенню задачі проєктування схем буферів даних. Розглянуто різні відомі методи побудови буферних схем. Серед них відмічені як перспективні методи відображення графу синхронних потоків даних (ГСПД) у буферну схему та методи побудови систолічних процесорів шляхом відображення графу потоку даних представленого у багатовимірному просторі. Запропоновано новий метод синтезу буферних схем на основі відображення ГСПД представленого у багатовимірному просторі, тобто, просторового ГСПД. На першому кроці методу будується ГСПД у трьохвимірному просторі з координатами номеру процесорного елементу (ПЕ), типу операції і номеру такту виконання операції. На другому етапі ГСПД урівноважується шляхом додавання вершин затримки у дуги, після чого він оптимізується перестановкою вершин у просторі з додержанням відповідних евристик. На третьому етапі ГСПД описується мовою VHDL. При цьому вершини затримки відображаються у ПЕ типу регістр. В результаті, одержується опис пристрою з буфером на основі ОЗП або конвеєра з регістрів в залежності від прийнятої евристики оптимізації. Окремо розглянута евристика, результатом застосування якої є використання таких апаратних примітивів у ПЛІС, як SRL16. Застосування методу показано на прикладі проєктування буферної схеми для процесора дискретного косинусного перетворення, результатом якого виявились проєкти буферів на базі примітивів SRL16 з екстремально малими апаратними витратами. Запропонований метод вбудовано в експериментальний фреймворк SDFCAD, призначений для синтезу конвеєрних операційних пристроїв для ПЛІС.

***Keywords**: FPGA, VHDL, synchronous dataflow, datapath synthesis*.

УДК 004.056.5
DOI: https://doi.org/10.20535/2708-4930.3.2022.265480

ОРГАНІЗАЦІЯ ШВИДКОГО ЕКСПОНЕНЦІЮВАННЯ НА ПОЛЯХ ГАЛУА ДЛЯ СИСТЕМ КРИПТОГРАФІЧНОГО ЗАХИСТУ ДАНИХ

(Стор. 17 – 25)

Аль-Мрият Гассан Абдель Жалил Халіл
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0002-4382-5309

Олександр Марковський
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0003-3483-4233

Іван Мозговий
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0001-5469-486X

Альона Ступак
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0002-3491-7365

Об'єктом викладених в статті досліджень, є процеси обчислення експоненти на кінцевих полях Галуа при реалізації криптографічних механізмів захисту інформації з відкритим ключем.

Мета цих досліджень полягає в прискоренні виконання операції експоненціювання на полях Галуа, яка є базовою для реалізації широкого кола криптографічних протоколів захисту даних за рахунок використання передобчислень, які залежать тільки від утворюючого поліному поля Галуа.

Для досягнення поставленої мети використана особливість виконання експоненціювання на полях Галуа в криптографії з відкритим ключем – сталість утворюючого поліному поля Галуа, який є частиною відкритого ключа. Це дозволяє виділити обчислення, що залежать лише від утворюючого поліному і виконати їх лише один раз зі збереженням результатів в таблицях передобчислень. Використання передобчислень дозволяє не тільки зменшити обчислювальну складність операції експоненціювання на полях Галуа, але й ефективно застосовувати для прискорення суміщення обробки декількох розрядів.

На основі цього в статті запропоновано організацію прискореного виконання базової операції широкого кола криптографічних алгоритмів з відкритим ключем – експоненціювання на кінцевих полях Галуа GF (2^n). Прискорення обчислювальної реалізації цієї операції досягається за рахунок організації обробки відразу декількох розрядів коду при піднесенні до квадрату на полях Галуа. Ця організація базується на використанні властивостей поліноміального квадрату, груповій редукції Монтгомері та широкому використанні передобчислень. Детально розроблені процедури виконання базових операцій експоненціювання на полях Галуа, робота яких ілюстрована числовими прикладами. Теоретично та експериментально доведено, що запропонована організація дозволяє прискорити обчислювальну організацію цієї важливої для криптографічних застосувань операції в 2,4 раз.

Ключові слова: *мультиплікативні операції на полях Галуа, криптографічні алгоритми, що базуються алгебрі полів Галуа, експоненціювання на полях Галуа, редукція Монтгомері*

УДК 004.056.5
DOI: https://doi.org/10.20535/2708-4930.3.2022.265418

ОРГАНІЗАЦІЯ ПАРАЛЕЛЬНОГО ВИКОНАННЯ МОДУЛЯРНОГО МНОЖЕННЯ ДЛЯ ПРИСКОРЕННЯ ОБЧИСЛЮВАЛЬНОЇ РЕАЛІЗАЦІЇ КРИПТОГРАФІЇ З ВІДКРИТИМ КЛЮЧЕМ

(Стор. 26 – 32)

Ігор Бояршин
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0002-5318-8234

Олександр Марковський
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0003-3483-4233

Богдана Островська
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0001-7967-4582

Об'єктом досліджень, яким присвячена стаття, є процеси обчислення мультиплікативних операцій модулярної арифметики, які виконуються над числами, довжина яких на порядки перевищує розрядність процесорів.

Ціль досліджень полягає в прискоренні виконання важливої для криптографічних застосувань операції модулярного множення над числами, розрядність яких значно перевищує розрядність процесора, за рахунок організації паралельного обчислення фрагментів модулярного добутку на багатоядерних комп'ютерах.

В якості основного шляху досягнення поставленої мети в представлених статею дослідженнях використано розпаралелювання на рівні обробки бітів множника та застосування групової редукції Монтгомері з використанням передобчислень, що залежать лише від модуля, котрий для криптографічних застосувань є частиною відкритого ключа, що дозволяє вважати його сталим.

У статті теоретично обґрунтовано, розроблено та досліджено спосіб паралельного виконання базової операції криптографії з відкритим ключем – модулярного множення чисел великої розрядності. В основу покладено спеціальну організацію поділу складових модулярного множення за незалежними обчислювальними процесами з тим, щоб забезпечити можливість ефективної групової редукції добутку. Запропонована організація забезпечує високу незалежність часткових обчислювальних процесів, що спрощує організацію взаємодії між ними. Для реалізації групової редукції Монтгомері передбачено використання результатів передобчислень, які залежать тільки від модуля і, відповідно, виконуються лише один раз. Виклад ілюструється числовими прикладами. Теоретично та експериментально доведено, що запропонований підхід до розпаралелювання обчислювального процесу модулярного множення з використанням групової редукції Монтгомері при використанні s процесорних ядер дозволяє прискорити цю важливу для криптографічних застосувань операцію в 0,57*s* раз.

Ключові слова: *модулярне множення, модулярна редукція Монтгомері, криптографія з відкритим ключем, паралельні обчислення, мультиплікативні операції модулярної арифметики.*

УДК: 004
DOI: https://doi.org/10.20535/2708-4930.3.2022.267665

МОДЕЛЮВАННЯ РУХУ РІДИНИ В ЗАКРИТИХ ПОВЕРХНЯХ ЗА ДОПОМОГОЮ РЕШІТЧАСТОЇ МОДЕЛІ БОЛЬЦМАНА

(Стор. 33 – 41)

Кузьмич Валентин
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0002-6077-3609
Новотарський Михайло

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0002-5653-8518

Відновлювальні операції на травному тракті людини можуть викликати негативні наслідки. Ці ефекти проявились у появі небажаних деформацій, так званих «сліпих мішків», які виникли внаслідок утворення зон високого тиску після зміни геометрії порожнини травного тракту під час реконструктивної операції. З цієї причини розробка математичної моделі руху рідин в закритих поверхнях в останні роки стала необхідною.

Існує багато підходів для розв'язування таких задач. Більшість традиційних підходів потребує значних затрат часу та обчислювальних ресурсів для їх реалізації. Так, при застосування аналітичних методів існують можливі розв'язки тільки за певних умов. Тому виникає необхідність використання ефективних чисельних методів, одним з яких є решітчаста модель Больцмана. Метою даної роботи є дослідження гідродинамічних процесів у замкнутих поверхнях за допомогою решітчастої моделі Больцмана.

Існує значна кількість публікацій присвячених моделюванню руху рідин в закритих поверхнях. Але аналіз публікацій показав, що використання решітчастої моделі Больцмана в таких задачах не є детально дослідженим.

Сформульована постановка задачі у вигляді чисельного рішення рівняння Больцмана. Запропоновано розв'язувати таку задачу шляхом використання решітчастої моделі Больцмана.

Результати теоретичних досліджень та експериментів показили практична доцільність використання запропонованого підходу до моделювання руху рідин в закритих поверхнях. Відмічено, що запропонований підхід має перспективи до подальшого дослідження та розвитку.

Key words: *решітчаста модель Больцмана, гідродинаміка*

УДК 0004.052.42
DOI: https://doi.org/10.20535/2708-4930.3.2022.269112

ІДЕНТИФІКАЦІЯ ВІДДАЛЕНИХ КОРИСТУВАЧІВ З НУЛЬОВИМ РОЗГОЛОШЕННЯМ З ВИКОРИСТАННЯМ ПСЕВДОВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ

(Стор. 42 – 48)

Ігор Дайко
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0002-5316-7080

Віктор Селіванов
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0001-8519-6038

Мирослава Чернишевич
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0002-5033-3028

Олександр Марковський
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0003-3483-423

Об'єктом викладених в статті досліджень є процеси криптографічно строгої ідентифікації учасників віддаленої інформаційної взаємодії, що забезпечують можливість захисту від перехоплення сеансу сторонніми особами.

Мета досліджень полягає в підвищенні ефективності криптографічно строгої ідентифікації учасників віддаленої інформаційної взаємодії за рахунок прискорення процесу підтвердження ідентичності, а також шляхом організації вторинних циклів контролю контакту для протидії перехоплення взаємодії.

Поставлена мета досягається за рахунок додаткового використання вторинних циклів ідентифікації, які періодично проводяться протягом сеансу взаємодії і дозволяють виявити факт перехоплення взаємодії зловмисником. Для реалізації основної та вторинної ідентифікації використовується єдиний криптографічний механізм – генератори псевдовипадкових двійкових послідовностей. Крім реалізації криптографічно строгої ідентифікації цей механізм може використовуватися для швидкого потокового шифрування даних, якими обмінюються учасники взаємодії.

В статті теоретично обґрунтовано, детально розроблено та досліджено схему ідентифікації на основі концепції «нульових знань» з використанням незворотних генераторів псевдовипадкових бітових послідовностей. Сеансові паролі утворюють ланцюжок, сформований вибірковими значеннями послідовності. Для протидії атакам з відтисненням однієї із сторін віддаленої взаємодії в запропонованій схемі передбачені сеанси вторинної ідентифікації. Детально розроблені основні елементи запропонованої схеми ідентифікації: процедури авторизації, первинної та вторинної ідентифікації.

Теоретично доведено, що задача злому запропонованого методу криптографічно строгої ідентифікації ідентична передбаченню двійкової послідовності, яку формує генератор. Для стандартизованих криптографічних генераторів псевдовипадкових послідовностей вирішення цієї задачі виходить для більшості практичних застосувань за межі технічних можливостей. Теоретично та експериментально доведено, що запропонована схема криптографічно строгої ідентифікації забезпечує на 2 – 3 порядки більшу швидкодію в порівнянні з відомими схемами, що використовують незворотні перетворення теорії чисел і на порядок швидша за схеми ідентифікації на основі ланцюжка хеш-перетворень.

***Ключові слова**: ідентифікація на основі концепції “нульових знань”, ланцюжки сеансових паролів, генератори псевдовипадкових бітових послідовностей, серединні атаки.*

УДК 004.052.42

DOI: https://doi.org/10.20535/2708-4930.3.2022.269132

ОРГАНІЗАЦІЯ ЗАХИЩЕНОЇ ФІЛЬТРАЦІЇ ЗОБРАЖЕНЬ В ХМАРАХ

(Стор. 49 – 55)

Аліреза Міратаеї
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0002-4732-7030

Русанова Ольга
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0003-3511-4438

Трибунська Кароліна
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0001-8268-2372

Олександр Марковський
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0003-3483-4233

Об’єктом дослідження є процеси гомоморфного шифрування зображень для їх захищеної середньоарифметичної фільтрації в хмарах.

Метою роботи є підвищення ефективності захищеної обробки зображень в хмарах, зокрема їх середньоарифметичної фільтрації на віддалених комп’ютерних системах за рахунок підвищення рівня захищеності.

В статті запропоновано підхід до використання хмарних технологій для прискорення фільтрації потоків зображень при забезпеченні їх захисту під час обробки на віддалених комп’ютерних системах. Гомоморфне шифрування зображень при їх віддаленій фільтрації пропонується здійснювати шляхом перемішування рядків матриць пікселів.

Вказаний підхід конкретизовано у вигляді методу гомоморфного шифрування зображень для захисту від їх незаконної реконструкції під час середньоарифметичної фільтрації на віддалених комп’ютерних системах який відрізняється тим, що в якості основного елементу захисту застосовано перемішування рядків матриці пікселів зображень. Порядок перемішування може обратися випадковим чином і виконує функції секретного ключа гомоморфного шифрування зображень. В рамках розробленого методу визначено процедури часткової середньоарифметичної фільтрації, яка здійснюється на віддалених системах, а також процедури фінальної стадії фільтрації, яка виконується на термінальній платформі, яка здійснює обробку та аналіз реального зображення. Розроблений метод захищеної фільтрації на основі перемішування рядків матриці пікселів дозволяє, за рахунок використання віддалених обчислювальних потужностей, прискорити цю операцію на 1 – 2 порядки, що практично збігається з аналогічними показником найбільш швидкодіючого варіанту захисту зобра¬жень на основі адитивного маскування.

Основна перевага розробленого методу полягає в значно більш більш високому рівні захищеності від спроб, з використанням статистичного аналізу, отримати незаконний доступ до зображень під час їх обробки на непідконтрольних користувачу віддалених комп’ютерних системах.

Запропонований метод може бути використаний для прискорення обробки та аналізу зображень термінальними пристроями комп’ютерних систем віддаленого моніторингу стану об’єктів реального світу і управління ними.

Ключові слова*: середньоамрифметична фільтрація, обробка зображень, гомоморфне шифрування, захищена обробка даних в хмарах*

УДК 004.056.5
DOI: https://doi.org/10.20535/2708-4930.3.2022.265479

ШВИДКЕ ЗАХИЩЕНЕ ОБЧИСЛЕННЯ В ХМАРІ ПРОЦЕДУР КРИПТОГРАФІЇ З ВІДКРИТИМ КЛЮЧЕМ ДЛЯ IoT

(Стор. 56 – 62)

Аліреза Міратаеї
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0002-4732-7030

Гайдукевич Марія
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0003-2334-2401

Олександр Марковський
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0003-3483-4233

Об'єктом викладених в статті досліджень є процеси захищеної реалізації на віддалених комп’ютерних системах базової операції криптографії з відкритим ключем – модулярного експоненціювання.

Мета дослідження полягає в прискоренні обчислювальної реалізації механізмів криптографічного захисту з відкритим ключем на термінальних мікроконтролерах комп’ютерних систем керування в режимі реального часу шляхом організації захищеного виконання базової операції цих механізмів – модулярного експоненціювання на віддалених комп’ютерних системах.

Для досягнення поставленої мети використано мультиплікативно-адитивне розкладення коду експоненти, яке дозволило розділити обчислення на дві частини, більша з яких виконується на віддалених комп’ютерних системах з використанням хмарних технологій, а менша за обсягом – на термінальному мікроконтролері. При цьому за даними, що передаються в хмару на віддалені комп’ютерні системи практично неможливо відновити секретні компоненти операції.

В результаті провдених досліджень теоретично обґрунтовано та розроблено метод прискорення реалізації на вбудованих термінальних мікроконтролерах *IoT* криптографічних механізмів захисту даних, базовою операцією яких є модулярне експоненціювання чисел великої розрядності. Метод ґрунтується на використанні для прискорення обчислень віддалених комп'ютерних систем та передбачає захист від реконструкції секретних ключів криптосистем за даними, що передаються в хмару. Основна відмінність запропонованого методу полягає в використанні єдиного механізму захисту секретних компонентів операції у вигляді мультиплікативного розкладення коду експоненти.

Теоретично та експериментально доведено, що метод дозволяє в середньому в 50 разів прискорити виконання протоколів криптографічного захисту даних у *IoT* при забезпеченні достатнього для більшості практичних застосувань рівня захищеності.

Ключові слова: *модулярне експоненціювання, захищене обчислення в хмарі, безпека IoT, криптосистеми з відкритим ключем.*

УДК 004.02, 004.2
DOI: https://doi.org/10.20535/2708-4930.3.2022.265229

МЕТОДИ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ МАСШТАБОВАНИХ СИСТЕМ: ОГЛЯД

(Стор. 63 – 76)

Гончаренко Олександр
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0002-9086-6988

Луцький Георгій
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0002-3155-8301

В статті розглянута проблема неефективності сучасних систем та горизонтального масштабування як методу збільшення продуктивності. Висвітлено основні проблеми, що складають собою згадану проблематику – обмеження паралелізму, невідповідність між задачею та системою, складнощі програмування та питання балансу між витратами та продуктивністю. Запропоновано класифікацію для можливих рішень, за якою їх було поділено на архітектурні та мережеві, та проведено огляд можливих рішень. В рамках архітектурного класу оглянуто такі підходи як квантові обчислення та парадигма *dataflow*, докладно проаналізовано найбільш перспективні підходи в їх межах.

Порівняльний аналіз показує, що за своєю природою *dataflow* та квантові процесори не протирічать одне одному, більш того – доповнюють в контексті проблематики. Так, спеціалізовані квантові обчислювачі *D-Wave*, на противагу універсальним квантовим

процесорам, надають великі обчислювальні потужності за відносно скромною ціною, в той час як dataflow рішення, представлене, переважно, процесорами Maxeler, є універсальними і ефективними, проте поступаються квантовим системам в ряді задач.

Обидва типи процесорів при цьому потребують певної мережі для зв'язку, що робить питання топології актуальним. На мережевому рівні розглянуто 2 топології – *FatTree* та *Dragonfly*, та виділено їх основні властивості. Аналіз показав, що в контексті проблематики *Dragonfly* є трошки кращою завдяки децентралізації та меншому діаметру, проте обидва рішення забезпечують гарні топологічні характеристики та підтримку основних сучасних технологій маршрутизації.

У висновках зазначено основні аспекти постановки проблеми та огляду, розглянуто подальші перспективи та можливі методи. В першу чергу, багатообіцяючою ідеєю є поєднання в одній системі квантових та неквантових рішень. Такий підхід дозволяє суттєво прискорити певні обчислення, при цьому забезпечуючи універсальність системи. Проте більш загальним питанням є взаємоінтеграція рішень як така. Проблема ефективності має багато часткових рішень, проте не всі вони є поєднуваними, тож, розробка комплексних методів на основі вже відомих є ключовою перспективою предметної області.

***Ключові слова**: підвищення ефективності, масштабовані системи, високопродуктивні обчислення, архітектура, топологія*

УДК 004.056.5
DOI: https://doi.org/10.20535/2708-4930.3.2022.266391

СУЧАСНІ ЗАСОБИ БЕЗПЕКИ ІНФОРМАЦІЙНИХ СИСТЕМ

(Стор. 77 – 86)

Клименко Ірина
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0001-5345-8806

Вернер Анна
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0001-8598-363X

Високі темпи технічного прогресу та поширення інформаційних технологій є досить розповсюдженим явищем сьогодення. Однак, статистичні дані вказують на те, що одночасно з позитивною динамікою спостерігається також і щорічне експоненційне зростання обсягу шкідливого програмного забезпечення, впливам якого піддаються інформаційні системи. Так у другому кварталі 2022 року системами безпеки було виявлено 55,3 млн шкідливих і потенційно небажаних об'єктів, що стали серйозною загрозою інформаційній безпеці, приймаючи різноманітні форми, включаючи атаки на програмне забезпечення, крадіжки інтелектуальної власності, крадіжки особистих даних, крадіжки інформації, саботаж та вимагання інформації. Саме тому технології аналізу та виявлення потенційних небезпек постійно вдосконалюються. Однак, наразі жоден метод не здатен виявити увесь існуючий спектр шкідливого програмного забезпечення, що засвідчує складність та необхідність створення ефективних підходів до виявлення шкідливого програмного забезпечення та про наявність необмеженого простору можливостей з розробки нових методів в даній галузі.

В даній статті виконано огляд фактичного стану інформаційної безпеки, класифікуються та висвітлюються специфічні атрибути механізмів безпеки, аналізуються різні критерії класифікації ризиків безпеки інформаційної системи.

В першому розділі розглянуто категорії та особливості видів загроз інформаційній безпеці. В другому розділі міститься загальний опис методів аналізу загроз, порівнюються статичний, динамічний та гібридний методи аналізу шкідливого програмного забезпечення та висвітлюються переваги і недоліки кожного з них. В третьому розділі розглядаються новітні засоби виявлення та протидії загрозам інформаційних систем, аналізуються особливості їх впровадження. У статті подано ретельний огляд поточних досліджень методології виявлення шкідливого програмного забезпечення

Метою даної статті є надання загального уявлення про сучасний стан інформаційної безпеки та існуючі сучасні методи захисту інформаційних систем від можливих загроз.

Ключові слова: *інформаційна безпека, інформаційні системи, засоби захисту, шкідливі програми*

УДК 004.02, 004.2

DOI: https://doi.org/10.20535/2708-4930.3.2022.265200

ОГЛЯД ІНСТРУМЕНТІВ OCR ДЛЯ ЗАВДАНЬ РОЗПІЗНАВАННЯ ТАБЛИЦЬ І ГРАФІКІВ У ДОКУМЕНТАХ

(Стор. 87 – 94)

Ярошенко Олександр
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/ 0000-0003-1871-3810

У цьому дослідженні представлено огляд інструментів *OCR* (оптичне розпізнавання символів) для розпізнавання таблиць документів і графіків. Оцифрування паперових документів має багато переваг як для фізичних осіб, так і для компаній. Для оцифрування потрібно використовувати програмне забезпечення *OCR*. Таке програмне забезпечення сканує документи, щоб зробити текст зрозумілим для комп'ютера. Їх можна конвертувати у формати, які підтримуються *Microsoft Word або Google Docs*. Програмне забезпечення *OCR* стає радше необхідністю, ніж утилітою для розваг. *OCR* створює текст із можливістю пошуку та редагування з друкованих документів, а також із відсканованих фотографій або книг і *PDF*-файлів.

Зараз спостерігається активна тенденція до цифровізації документів. Існує великий попит на рішення, які можуть ефективно автоматизувати обробку великого масиву документів з високою точністю. Окремим випадком є обробка *PDF*-файлів, таких як відскановані документи або створені програмними редакторами. Рішення *OCR* спрямовані на підвищення ефективності обробки та аналізу цифрових документів за допомогою штучного інтелекту. Цими рішеннями можуть користуватися як державні установи, так і підприємства. Розроблені системи можуть стати цінним доповненням до *CRM*-систем і можуть бути інтегровані замість існуючих модулів обробки документів або використовуватися як окреме рішення.

Хоча існуючі рішення *OCR* можуть ефективно розпізнавати текст, розпізнавання графічних елементів, таких як діаграми та таблиці, все ще знаходиться на стадії розробки. Рішення, які можуть підвищити точність розпізнавання візуальних даних, можуть бути цінними для обробки технічних документів, таких як наукові, фінансові та інші аналітичні документи.

Ключові слова: *OCR, файли PDF, FastText, виявлення, розпізнавання, глибоке навчання, технічні документи.*