The development of Internet technologies together with mobile and computer technologies have formed smart technologies that allow the formation of both cyber-physical and socio-cyber-physical systems. The basis of smart technologies is the integration of wireless channel standards with mobile and computer protocols. 4G/5G technologies are integrated with various web platforms, taking into account the digitalization of services in cyberspace. But the SSL/TLS protocol, based on the hybridization of symmetric encryption algorithms with hashing algorithms (AEAD mode), which is supposed to provide security services, is vulnerable to "Meet in the middle", POODLE, BEAST, CRIME, BREACH attacks. In addition, with the advent of a full-scale quantum computer, symmetric and asymmetric cryptography algorithms that provide security services can also be hacked. To increase the level of security, an improved protocol based on post-quantum algorithms – crypto-code constructions is proposed, which will ensure not only resistance to current attacks, but also stability in the post-quantum period. To ensure the "hybridity" of services, it is proposed to use the McEliece and Niederreiter crypto-code constructions (confidentiality and integrity are ensured) and the improved UMAC algorithm on the McEliece crypto-code construction. Taking into account the level of "secrecy" of information, it is suggested to use various combinations of crypto-code constructions on different algebraic geometric and/or flawed codes. The use of crypto-code constructions not only provides resistance to attacks, but also simplifies the formation of a connection – the parameters of elliptic curves are used to transmit a common key. This approach significantly reduces the connection time of mobile gadgets and simplifies the procedure of agreement before data transfer

Keywords: improved SSL/TLS protocol, post-quantum encryption algorithms, improved UMAC algorithm, algebraic geometric codes, flawed codes

UDC 681.32:007.5
DOI: 10.15587/1729-4061.2023.281795

DEVELOPMENT OF AN IMPROVED SSL/TLS PROTOCOL USING POST-QUANTUM ALGORITHMS

Serhii Yevseiev
Corresponding author
Doctor of Technical Sciences, Professor, Head of Department*
E-mail: Serhii.Yevseiev@gmail.com
Alla Havrylova
Senior Lecturer *
Stanislav Milevskyi
PhD, Associate Professor *
Igor Sinitsyn
Doctor of Technical Sciences, Senior Researcher
Institute of Software Systems of the National Academy of Sciences of Ukraine
Akademika Hlushkova ave., 40, Kyiv, Ukraine, 03187
Volodymyr Chalapko
Faculty of Weapons and Military Equipment
Military Institute for Tank Troops
Poltavsky Shlyakh str., 192, Kharkiv, Ukraine, 61198
Hennady Dukin
PhD, Associate Professor
Department of Aviation Radiotechnical Systems of Navigation and Landing
Ivan Kozhedub Kharkiv National Air Force University
Sumska str., 77/79, Kharkiv, Ukraine, 61003
Vitalii Hrebeniuk
Doctor of Law, Senior Reseacher, First Vice-Rector
National Academy of the Security Service of Ukraine
Mykhaila Maksymovycha str., 22, Kyiv, Ukraine, 03066
Mykhailo Diedov
Scientific-Research Institute of Military Intelligence
Yuriia Illienka str., 81, Kyiv, Ukraine, 04050
Lala Bekirova
Doctor of Technical Science, Professor, Head of Department
Department of Instrumentation Engineering
Azerbaijan State Oil and Industry University
Azadliq ave., 20, Baku, Azerbaijan, AZ 1010
Oleksandr Shpak
PhD, Associate Professor
Department of Software Systems
Uzhhorod National University
Narodna sq., 3, Uzhhorod, Ukraine, 88000
*Department of Cyber Security
National Technical University "Kharkiv Polytechnic Institute"
Kyrpychova str., 2, Kharkiv, Ukraine, 61002

Received date 05.04.2023
Accepted date 15.06.2023
Published date 30.06.2023

***How to Cite**: Yevseiev, S., Havrylova, A., Milevskyi, S., Sinitsyn, I., Chalapko, V., Dukin, H., Hrebeniuk, V., Diedov, M., Bekirova, L., Shpak, O. (2023). Development of an improved ssl/tls protocol using post-quantum algorithms. Eastern-European Journal of Enterprise Technologies, 3 (9 (123)), 33–48. doi: https://doi.org/10.15587/1729-4061.2023.281795*

1. Introduction

The development of modern information and communication technologies, mobile technologies and the Internet of things has significantly expanded the range of digital services and formed a trend in the formation of cyber-physical systems based on the integration of these technologies with cyberspace. The growth in the capabilities of mobile Internet technologies and wireless channels makes it possible to form mesh networks and

expand the areas of smart technologies. Such systems typically use the Secure Sockets Layer (SSL)/Transport Level Security (TLS) (SSL/TLS) protocol stack to provide security services, which allows providing authentication, integrity, and confidentiality services [1–5]. This stack is formed on complex symmetric algorithms that simultaneously provide both encryption and generation of MAC codes [1, 6–8]. When this protocol stack is used in smart technologies, it can be combined with Wi-Fi Protected Access (WPA) Enterprise to protect corporate-level wireless networks [7, 8]. At the transport layer in cyber-physical systems, the TLS protocol provides tunnel mode during the authentication process. In addition, at the transport level, the SL/TLS stack is used in public key infrastructure (PKI) technologies, provides transparent data transfer between end users, and also provides reliable data transfer to higher levels [4, 6, 9–11]. SSL cryptographic protocols and TLS transport layer security provide data encryption and authentication between various homogeneous and heterogeneous devices such as servers, computers and software applications running across the network. In smart technologies, the use of the SSL/TLS protocol stack is integrated with blockchain technology, as well as with the EDS Elliptic Curve Digital Signature Algorithm (ECDSA) based on elliptic curves, which will ensure system security until the post-quantum period [3, 10, 12–16]. However, the use of such an approach is associated with an increase in the energy intensity of the protocol and the need to take into account the hardware characteristics of the physical environment of cyber-physical systems [3, 7, 9]. In addition, the authors of [3, 5, 7, 8, 11, 14, 15] indicate significant problems with ensuring the required level of security when using this protocol stack. This is due to the possibility of implementing a number of attacks (Heartbleed, Man in the middle (MITM) and Bleichenbacher) based on the "handshake" algorithm when exchanging control information between the client and the server. In addition, the emergence of a full-scale quantum computer and the possibility of implementing Grover's (breaking symmetric algorithms) and Shor's (breaking asymmetric cryptography algorithms) quantum algorithms require new solutions based on post-quantum algorithms, among which a group of algorithms based on methods and mathematical apparatus of the theory of error-correcting coding and/or based on Galois field theory. Among such algorithms, the McEliece and Niederreiter crypto-code constructions can also be used. Thus, it is necessary to ensure not only an increase in the level of resistance and the "elimination" of the possibility of implementing threats when using this protocol ("handshake" mode), but also its use in the post-quantum cryptoperiod.

2. Literature review and problem statement

The analysis [7, 8, 14, 15] showed that, despite the widespread use of the SSL/TLS v.1.3 protocol, in almost all cyberspace technologies, this protocol is subject to attacks that allow access to transmitted information. Thus, [7] considers threats to the TLS protocol and, as a countermeasure, proposes to use TLS-Monitor, a network monitoring tool that supports TLS attacks, which checks the traffic of the target system in search of vulnerabilities. However, this approach only allows you to control and does not allow you to eliminate the causes of attacks. In [8], the authors argue that wireless networks are prone to a man-in-the-middle SSL/TLS attack or an evil twin attack, when a requestor connects and unwittingly sends authentication credentials to a fake access point. These vulnerabilities are a consequence of imperfect designs and implementation of WPA software modules, which significantly affects the level of security in smart technologies. However, no effective protection measures are proposed. In [14], it is noted that SSL (Secure Sockets Layer) certificates have become a major part of website security. These certificates encrypt all communications with the website's public key and decrypt them with the certificate's private key, which is stored on the server. However, the exchange of control information before establishing a connection "provides" the ability to both intercept and "insert" a backdoor into the control packets of the "handshake" algorithm. In addition, the round-trip time (0-RTT) algorithm, which provides a "return" to the session without checking key data, can also be used to implement hacking of this connection. It is noted in [15] that the SSL/TLS protocol has a strong time dependence. Thus, the process of transition of the protocol state from the handshake phase to the information transfer phase requires not only taking into account information about the time context, but it is also necessary to take into account the presentation of data with a high level of separation. The authors propose to monitor this spectrum to assess possible vulnerabilities and threats, which will allow timely identification of the noted vulnerabilities and threats. However, this also does not eliminate the causes of their occurrence. In addition, the emergence of new vulnerabilities and/or threats may be "overlooked" by the proposed approach; there are no economic calculations to justify the cost-effectiveness of the solution. In [3], the authors propose to use post-quantum algorithms to ensure the security of the handshake algorithm. The paper presents the results of research on handshake performance using two devices with limited resources based on classical encryption and digital signature schemes, as well as on the basis of post-quantum standards. However, the results of the experiment showed that post-quantum algorithms have additional overhead for messages, which does not allow using them in smart technologies. The paper [5] considers the use of the SSL/TLS protocol in social networks and instant messengers. In particular, the communication flow of WhatsApp Web at the network layer can be examined using network traffic. After adding end-to-end encryption, WhatsApp updated its network security protocols to be intrusion-resistant. However, the authors argue that information is possible that can be obtained by examining the network traffic of the WhatsApp web application between the sender and the recipient. In [6], to eliminate SSL/TLS connection vulnerabilities, it is proposed to use readily available Internet certificates as initial certificates and use code coverage to direct the certificate change to create a set of various certificates. However, their use does not provide the required level of stability in the post-quantum period with a further increase in the volume of information flows. In [9], it is

proposed to reduce communication delay and processing of information flows in 4G/5G control technologies by uploading them to crypto externs (Crypto Extern) for Netronome Agilio smartNIC network cards. To provide confidentiality and authenticity services, it is proposed to use the ChaCha20 symmetrical streaming algorithm. However, smart network adapters have a limited set of instructions and limited memory, making it difficult to implement security algorithms. The presented evaluation shows that the proposed Crypto Extern implementation meets the scalability requirements of popular applications such as serverless control functions and in-band host telemetry [9]. However, ensuring the required level of security in the post-quantum crypto period is questioned. In addition, there are hardware limitations that would otherwise require sending an application message to the host VM/container for cryptographic operations, negating the benefits of offloading. In [10], to provide security services, it is proposed to use blockchain technology to solve problems related to the excessive power of the certification authority and revocation and request difficulties, as well as substitution of certificates in the handshake phase of the SSL/TLS protocol. To eliminate the drawback, the TS-PBFT algorithm is proposed based on the adaptation of the threshold signature technology to the Practical Byzantine Fault Tolerance (PBFT) algorithm. The TS-PBFT algorithm reduced communication overhead by reducing communication complexity, increased control by introducing an external monitoring mechanism into the selection of the presentation change protocol master node, and improved the performance of the consensus mechanism by adding a batch processing mechanism. However, this approach significantly increases energy consumption and allows attackers to access databases on any node of the system. In [11], it is proposed to use a comprehensive key management system (QKPT), which allows transferring user private keys (BYOPK) to multi-user clouds. QKPT presents a carefully designed key wrapper layer to overcome these issues. A small symmetric wrapper key (SWK) is generated for each tenant as a master key to solve the first two problems, while a special private key wrapping scheme is used to remove the transparency constraint. In addition, QKPT includes certificate trust to improve SWK lifecycle security and is compatible with the SSL/TLS protocol. However, cloud technologies are also subject to attacks, which calls into question the stability of this solution. It is indicated in [12] that network traffic is increasingly valued for privacy protection, and encrypted SSL/TLS (Secure Sockets Layer/Transport Layer Security) traffic is growing, but more and more malicious activity is hidden in it. Existing detection methods are less accurate at detecting new and unknown malicious traffic. The authors propose a method for detecting malicious SSL/TLS traffic based on adaptive feature learning. The model can automatically extract key classification information from untagged malicious encrypted SSL/TLS traffic. The model also uses 5-Tuple-Masking technology to optimize input data, which greatly improves the model's ability to adapt to new malicious traffic in complex network environments. However, given the rapid development of computing capabilities, the total accuracy reaches 89.25 %, which does not allow timely response to targeted threats. In order to provide the authenticity service, it is proposed in [13] to use a secure blind registration protocol without certificates (Cumulative Layout Shift, CLS-BPR) instead of certificates transmitted based on the SSL/TLS protocol. This protocol is a password registration scheme based on identity-based cryptography, i.e. both the user and the service provider are authenticated with their short-lived identity-based secret key. For secure storage, a bilinear map with salt is used, so in the case of an offline attack, the adversary is forced to calculate a computationally costly bilinear map for each password candidate and salt, which slows down the attack. However, the stability of the bilinear map itself is not determined, which casts doubt on the stability of the entire system. In [16], the authors suggest using the best SSL/TLS and elliptic curve encryption algorithms in e-commerce. However, the paper does not substantiate the costs of performance and energy intensity, which can significantly affect their use in mobile Internet technologies on wireless communication channels.

Thus, the analysis showed that in order to "eliminate" (minimize) existing threats to the SSL/TLS protocol, it is proposed to use either network traffic monitoring or post-quantum cryptography and elliptic curve cryptography algorithms. The first approach does not eliminate the reasons for the implementation of threats, and the second does not provide the required level of performance and use in smart technologies based on smart chips with limited computing resources.

3. The aim and objectives of the study

The aim of this study is to develop an improved SSL/TLS protocol based on post-quantum algorithms and an improved cascade hashing algorithm. This approach simplifies the handshake phase, does not require significant computing resources, and ensures the required level of security.

To achieve the aim of the work, the following objectives are accomplished:

– to develop complex algorithms for ensuring confidentiality, integrity and authenticity based on post-quantum algorithms for the SSL/TLS protocol;

– to justify the direction of improvement of the SSL/TLS protocol block diagram;

– to evaluate the energy costs for software implementation and the strength of the proposed hybrid cryptosystem.

4. Materials and methods of the study

4. 1. Object and hypothesis of the study

The object of the study is the process of ensuring security in cyber-physical systems based on post-quantum algorithms.

The hypothesis of the study was as follows. To ensure the security of mesh-, sensor network technologies using wireless channel standards: mobile technologies LTE (Long-Term Evolution), IEEE802.16 (WiMAX standard), IEEE802.16e (SOFDMA – Scalable OFDM

Access), IEEE802.15.4 (the standard is the basis for the ZigBee, WirelessHART, MiWi protocols), IEEE802.11 (Wi-Fi), Bluetooth, new approaches to providing security services are needed. In the context of the emergence of a quantum computer, it is necessary not only to use post-quantum cryptoalgorithms, but also a new approach to ensuring the security of socio-cyber-physical systems. Such algorithms require an increase in key sequences to 512 bits for symmetric cryptosystems (this provides a safe time of about 60 years), or the use of post-quantum asymmetric cryptosystems (PQAS). Among the contestants of the third round of the competition, algorithms built on the combination of the theory of error-correcting coding and cryptography stand out [17–23]. McEliece and Niederreiter crypto-code constructions on algebrogeometric codes (elliptic codes over the Galois field $GF(2^8)$, which provide protection against the Sidelnikov attack and reduce energy consumption. In addition, they provide integrated error correction in the information sequence [24]. Both crypto-code constructions are based on the principle of using the theory of error-correcting coding and orthogonality of the matrices G, the generating matrix of the linear code, and H, the check matrix of the linear code. As a key sequence in both crypto-code constructions, masking matrices are used: X, P, D. X is a masking non-degenerate random equiprobable matrix formed by a key source $k\times k$ with elements from $GF(q)$. P is a permutation random equiprobable matrix formed by a key source $n\times n$ with elements from $GF(q)$. D is a diagonal matrix formed by a key source $n\times n$ with elements from $GF(q)$; G – generating matrix with dimensions $k\times n$ (McEliece *CCC*); H – check matrix with dimension $r\times n$. In addition, a distinctive feature of Niederreiter's *CCC* is the preliminary use of equilibrium coding, which provides a practically relative coding rate equal to one. To reduce the computational costs and energy consumption of software implementation, without reducing the security level, it is proposed to use modified (shortened and/or extended elliptic codes) [17].

To build crypto-code constructions (CCC) based on modified (shortened or extended) cyclic codes on elliptic curves, the parameters presented in Table 1 are used. Table 2 shows the parameters of asymmetric cryptosystems on the corresponding CCC.

Table 1

Main (n, k, d) MEC parameters

Parameters	Shortened MEC	Extended MEC
(n, k, d) parameters of the code built through the mapping of the form $\varphi: X\rightarrow P^{k-1}$	$n=2\sqrt{q}+q+1-x$, $k\geq\alpha-x$, $d\geq n-\alpha$, $\alpha=3\times\deg F$, $k+d\geq n$	$n=2\sqrt{q}+q+1-x+x_1$, $k\geq\alpha-x+x_1$, $d\geq n-\alpha$, $\alpha=3\times\deg F$
(n, k, d) parameters of the code built through the mapping of the form $\varphi: X\rightarrow P^{r-1}$	$n=2\sqrt{q}+q+1-x$, $k\geq n-\alpha$, $d\geq\alpha$, $\alpha=3\times\deg F$, $k+d\geq n$	$n=2\sqrt{q}+q+1-x+x_1$, $k\geq n-\alpha$, $d\geq\alpha$, $\alpha=3\times\deg F$

Table 2

Main parameters of modified cryptosystems based on McEliece CCC on *MEC*

Parameters	Shortened MEC	Extended MEC
Dimension of the secret key	$l_{K+}=x\times\left\lfloor\log_2\left(2\sqrt{q}+q+1\right)\right\rfloor$	$l_{K+}=(x-x_1)\times\log_2\left(2\sqrt{q}+q+1\right)$
Dimension of the information vector	$l_I=(\alpha-x)\times m$	$l_I=(\alpha-x+x_1)\times m$
Cryptogram dimension	$l_S=\left(2\sqrt{q}+q+1-x\right)\times m$	$l_S=\left(2\sqrt{q}+q+1-x+x_1\right)\times m$
Relative encoding speed	$R=(\alpha-x)/\left(2\sqrt{q}+q+1-x\right)$	$R=(\alpha-x+x_1)/\left(2\sqrt{q}+q+1-x+x_1\right)$

When forming complex algorithms, taking into account the limitations of smart technologies, it is proposed to use the synthesis of post-quantum algorithms with multichannel cryptography on flawed codes. This approach provides the possibility of practical implementation without reducing the level of resistance on low-capacity chipsets in cyber-physical systems [17].

As an algorithm for generating a MAC code, it is proposed to use the fastest hashing algorithm, which is based on the cascade application of hash functions, the UMAC algorithm [18, 25–28]. The first layer of this hashing algorithm uses the UHASH universal hashing functions, which not only provide maximum hash code generation performance, but also allow estimating the number of collisions with uniform distribution over the entire set of hash codes. The number of collisions can be used as a universal identifier with a uniform distribution of hash codes throughout their set. To ensure the security of cascading hashing, the UMAC algorithm on the third layer uses bitwise addition of the resulting hash code with a pseudo-random pad (*Pad*). However, the use of the AES-256 block cipher as an algorithm for generating a pseudo-random sequence does not "preserve" the universality property.

To ensure authenticity in the post-quantum cryptoperiod, it is proposed to use the improved UMAC algorithm based on McEliece (Niederreiter) crypto-code structures [18, 28], which makes it possible to meet the requirements for the efficiency of processing information flows, "preserve" the universality property.

The scheme for transmitting a message from the sender to the recipient and checking the integrity of the received message by comparing the codegrams and hash codes using the McEliece CCC on modified elliptic codes (EC (MEC)) is shown in Fig. 1.

The algorithm for forming the pad is the McEliece's crypto-code construction on elliptic codes, MEC.

Applying modification changes to elliptic codes reduces the load on computing resources and leads to an increase in the efficiency of generating MAC codes in real time.

In [25], it was proposed to use the McEliece's crypto-code construction using flawed codes as a mechanism for forming a pseudo-pad of the third UMAC layer. The main ideas of flawed cryptography were proposed in [29, 30].

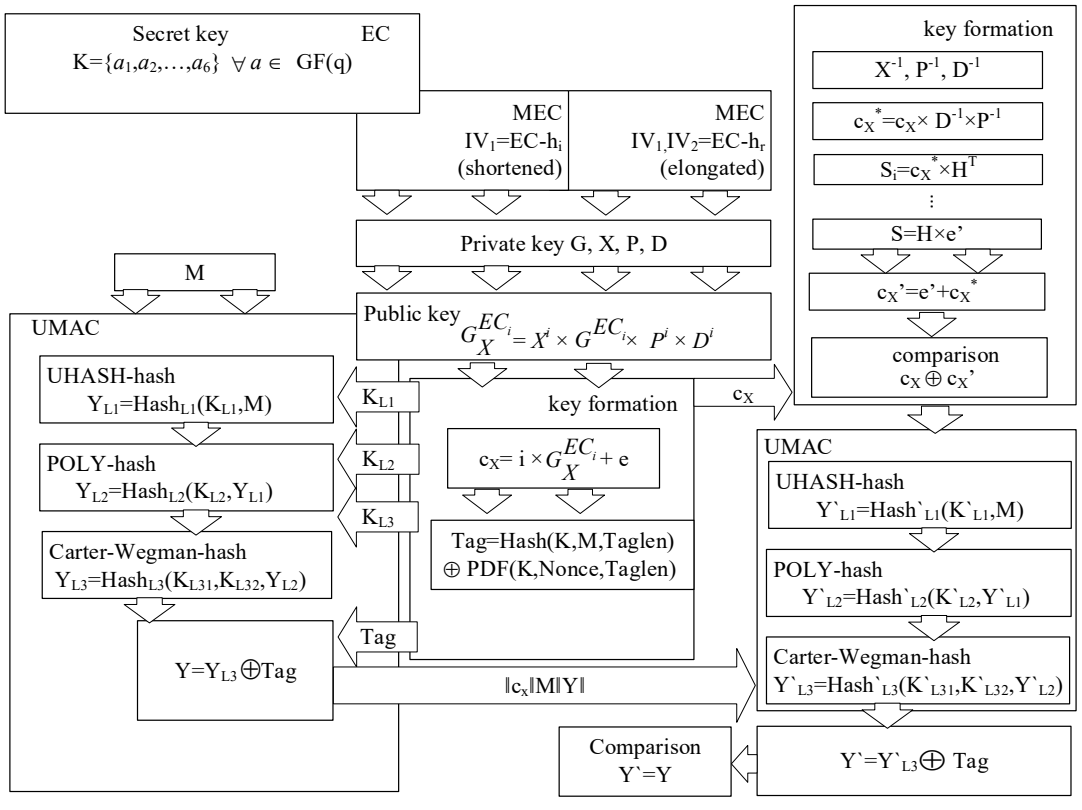


Fig. 1. Block diagram of the UMAC protocol based on the McEliece's crypto-code construction with modified (extended/shortened) codes

4. 2. Mathematical model of hash code generation based on the UMAC cascade algorithm

For building mathematical models to form a hash code of a transmitted message and a pseudo-random pad, the following input data are used:

M – transmitted plaintext;

I – plaintext information symbols (*k*-bit information vector over *GF*(*q*));

K – secret key;

Taglen – an integer from the set of valid values {4, 8, 12, 16} specifying the length of the message authentication code *Tag* in bytes;

Hash(*K, M, Taglen*) – the function of key universal hashing of an information message using the secret key *K*;

Y_{L1I} – the value of the universal hash function (UHASH-hash) of the first level of hashing;

Y_{L3I} – the value of the hash function (Carter-Wegman-hash) of the third level of hashing;

T – data block;

Blocklen – data block length (bytes);

Keylen – private key length (32 bytes);

Tag – integrity and authenticity control code;

K_{L1I} – secret key of the first level of hashing, consisting of subkeys $K_1, K_2, ..., K_n$;

K_{L3I} – secret key of the second level of hashing, consisting of keys K_{L31} (subkeys $K_1, K_2, ..., K_n$) and K_{L32} (subkeys $K_1, K_2, ..., K_n$);

Numbyte – pseudo-random key sequence length (number of subkeys);

K` – pseudo-random key sequence;

Index – number of subkey;

Wordbits [64, 128];

Maxwordrange – positive integer less than $2^{Wordbits}$;

k – an integer dependent on the key K_{L2} from the range [0,..., *prime*(*Wordbits*)–1], *prime* (*x*) – the largest prime number less than 2^x;

$M_P=Y_{L1}=Hash_{L1}(K_{L1},M)$ – data subject to polynomial hashing.

Based on the cascade representation of the UMAC algorithm, the mathematical model for generating the hash code of the transmitted message will consist of three levels.

The first level of the hash code splits the array-string *M* of up to 2^{64} bytes into blocks M_i of 1,024 bytes each, followed by the conversion of each block by the function $NH(K_{L1},M_i)$. The results obtained $Hash_{L1i}=NH(K_{L1},M_i)$ are concatenated (combined) into the string $Y_{L1}=Hash_{L1i}=NH(K_{L1},M_i)$, which is 128 times shorter than the information sequence. This string is the result of the first-level hashing:

$$Y_{L1} = Hash_{L1}(K_{L1},M) =$$
$$= NH(K_{L1},M_0)\|NH(K_{L1},M_1)\|...\|NH(K_{L1},M_{n-1}),$$

where

$$n=\left\lceil \frac{Length(M)}{1024} \right\rceil.$$

The value of the $Hash_{L1i}=NH(K_{L1},M_i)$ function is calculated according to the following rule. The information block M_i is split into four-byte subblocks so that $M_i = M_{i_1}\|M_{i_2}\|...\|M_{i_t}$, where $t=\left\lceil \frac{Length(M_i)}{4} \right\rceil$.

In this case, $t=\left\lceil \frac{1024}{4} \right\rceil = 256.$

Similarly, the key sequence K_{L1} is represented as sequences of four-byte subblocks:

$$K_{L1} = K_{L1_1} \| K_{L1_2} \| ... \| K_{L1_t}.$$

After that (taking the initial state $Hash_{L1i}$=0) for all j=1,9,17,..., t–7, the following operations are performed:

$$Hash_{L1_i} = Hash_{L1_i} + \\ +_{64}\left(\left(M_{i_{j+0}} +_{32} K_{L1_{j+0}}\right)\times_{64}\left(M_{i_{j+4}} +_{32} K_{L1_{j+4}}\right)\right),$$

$$Hash_{L1_i} = Hash_{L1_i} + \\ +_{64}\left(\left(M_{i_{j+1}} +_{32} K_{L1_{j+1}}\right)\times_{64}\left(M_{i_{j+5}} +_{32} K_{L1_{j+5}}\right)\right),$$

$$Hash_{L1_i} = Hash_{L1_i} + \\ +_{64}\left(\left(M_{i_{j+2}} +_{32} K_{L1_{j+2}}\right)\times_{64}\left(M_{i_{j+6}} +_{32} K_{L1_{j+6}}\right)\right),$$

$$Hash_{L1_i} = Hash_{L1_i} + \\ +_{64}\left(\left(M_{i_{j+3}} +_{32} K_{L1_{j+3}}\right)\times_{64}\left(M_{i_{j+7}} +_{32} K_{L1_{j+7}}\right)\right),$$

where $+_{64}$, $+_{32}$ – modulo 2^{64} and 2^{32} addition operations, respectively; $\times_{64}$ – modulo 2^{64} multiplication operation.

As a result of calculations, an eight-byte value of $Hash_{L1i}$ is formed.

The second level of the hash code uses a polynomial key transformation *Poly*. The result of this level is the hash code $Y_{L2} = Hash_{L2}(K_{L2}, Y_{L1}) = Poly(Wordbits, Maxwordrange, k, M_P)$, i. e. the string $Y_{L1} = Hash_{L1}(K_{L1}, M)$ is fed to the input of the second-level hashing.

As input data, the polynomial hash function uses.

According to the specification of the UMAC algorithm, the following constants are used as $prime(x)$: $prime(36)=2^{36}-5$, $prime(64)$ $2^{64}-59$, $prime(128)$ $2^{128}-159$. The bit length M_p is denoted by $Bytelength(M_P)$. Depending on the length M_p, the following features are used in the implementation of the second level of hashing:

– if the length of the received data M_p does not exceed 2^{17} bytes, then polynomial hashing *Poly* is performed with the parameters $Wordbits$=64; $Maxwordrange$ =$2^{64}-2^{32}$; k=$k64$ – the string formed by the first eight bytes of the key K_{L2} and a special eight-byte mask;

– if the length of the received data M_p exceeds 2^{17} bytes (but does not exceed 2^{64} bytes), then the first 2^{17} bytes of data are processed by the polynomial hash function $Poly(64, 2^{64}-2^{32}, k64, M_P)$, and the remaining data bytes are processed by the *Poly* function with the parameters $Wordbits$=128; $Maxwordrange$ =$2^{128}-2^{96}$; k=$k128$ – the string formed by the last 16 bytes of the key K_{L2} and a special 16 byte mask.

The hashed data M_p is split into blocks $Wordbytes$= =$Wordbits/8$ bytes:

$$M_P = M_{P_1} \| M_{P_2} \| ... \| M_{P_n},$$

where n=$Bytelength(MP)/Wordbytes$.

The result of hashing is the value of the polynomial function:

$$Y_{L2} = \left(M_{P_n} + kM_{P_{n-1}} + ... + k^{n-1}M_{P_1} + k^n\right) \bmod (p),$$

which is calculated by an iterative procedure (for all i=1,2,...,n):

$$Poly_i = \left(kPoly_{i-1} + M_{P_i}\right) \bmod (p),$$

$$Poly_0 = 1, p = prime(Wordbits),$$

using the Horner scheme:

$$M_{P_n} + kM_{P_{n-1}} + ... + k^{n-1}M_{P_1} + k^n = \\ = \left(\left(\left(k + M_{P_1}\right)k + M_{P_2}\right)k + ... + M_{P_{n-1}}\right)k + M_{P_n}.$$

The calculated hash value Y_{L2}=$Poly_n$ is an integer from the range [0,..., $prime$($Wordbits$)–1].

The third level of the hash code $Hash_{L3}(K_{L31}, K_{L32}, Y_{L2})$ is performed on the result of polynomial hashing and converts the data of length up to 16 bytes given to its input into the hash code Y of a fixed length of 32 bits.

The initial data of the third hashing level are two key sequences K_{L3_1} and K_{L3_2} with a length of 64 and 4 bytes, respectively, and the input 16 byte sequence Y_{L2}.

The hashed data Y_{L2} and the key sequence K_{L3_1} are evenly divided into eight blocks, each of which is represented as an integer Y_{L2_i} and $K_{L3_{1i}}$, i=1,2,...,8.

The hash value Y_{L3} is calculated as follows:

$$Y_{L3} = \left(\left(\left(\sum_{i=1}^{m} Y_{L2_i} K_{L3_{1i}}\right) \bmod \left(prime(36)\right)\right) \bmod \left(2^{32}\right)\right) xor \left(K_{L3_2}\right),$$

where $(x)xor(y)$ – XOR operation on x and y values.

4. 3. Mathematical model for the formation of a pseudo-random pad *Pad*

To form a pseudo-random pad *Pad*, McEliece crypto-code constructions (CCC) are used with the possibility of modifications (shortening or extending) (on MEC). This modified (shortened/extended) algebrogeometric (n, k, d)-code C_{k-h_j} (shortening)/ C_{h_r} (extending) with a fast decoding algorithm masquerades as a random (n, k, d)-code. $C_{k-h_j}{}^*$(shortening)/ $C_{h_r}{}^*$ (extending) is formed by multiplying the generating matrix G^{EC} – code C_{k-hj}/C_{hr} by the masking matrix, which are kept secret X^u, P^u and D^u – generation of the user's public key [17]:

$$G_X^{ECu} = X^u \times G^{EC} \times P^u \times D^u,\ u \in \{1,2,...,s\},$$

where G^{EC} – generating matrix of the algebrogeometric block (n, k, d)-code with elements from $GF(q)$ built on the basis of using user-selected polynomial coefficients of the curve $a_1...a_6$, $\forall a_i \in GF(q)$, which uniquely define a specific set of curve points from the space P^2.

Formation of the closed text $C_j \in C_{k-hj}$(shortening)/$C_j \in C_{hr}$ (extending) by the entered plaintext M and the given public key $G_{X\,a_i}^{ECu}$, $u \in \{1,2,...,s\}$ is carried out by forming a code word of the masked code with the addition of a randomly generated vector $e = (e_0, e_1, ..., e_{n-1}): C_j = \phi_u\left(M_i, G_X^u\right) = M_i \times \left(G_X^u\right)^T + e$, and the Hamming weight (the number of nonzero elements) of the vector e does not exceed the correcting ability of the algebraic block code used.

For each private text that is generated $C_j \in C_{k-hj}$ (shortening)/$C_j \in C_{hr}$ (extending), the corresponding vector e=(e_0,e_1,... e_{n-1}) is a one-time session key, that is, the vector e for a specific e_j is formed randomly, equally probable and independently of other closed texts. Thus, the generated cryptogram based on the post-quantum algorithm – McEliece crypto-code construction on modified elliptic codes is used as a pad. At the same time, the formation of CCC on MEC is provided over the $GF(2^6)$ field without reducing the level of cryptographic

strength. To "compensate" for the strength, initialization vectors are used, which define in MEC on shortened codes – shortening elements, and on extended codes – extending elements.

To further reduce computational costs, it is proposed to use cryptography on flawed codes. This approach makes it possible to reduce computational costs by 20 times without reducing security and form a pseudo-random pad when using CCC with MEC on flawed codes [29, 30].

Taking into account the modifications obtained during the formation of the codegram on modified (shortened or extended) codes, the damage is carried out according to the following scheme:

1. A subset of points $h(GF(q))$:$(P_{x1}, P_{x2}, ..., P_{xx})$, $h \subseteq EC(GF(q))$, $|h|=x$ is formed and kept secret.

The input for this is the final field $GF(q)$, elliptic curve $y^2z+a_1xyz+a_3yz^2=x^3+a_2x^2z+a_4xz+a_6z^3$, and also the set of its points $EC(GF(q))$: $(P_1, P_2, ..., P_N)$ over $GF(q)$.

2. An initialization vector is formed:

– when shortening characters – $IV_1=EC-h_j$, where IV – initialization vector;

– when extending characters – IV_1, $IV_2=EC-h_r$.

3. A codeword c is formed according to the entered information vector I. If the (n, k, d)-code over $GF(q)$ is given by its generating matrix, then $c=I\times G$.

4. A random error vector e is formed such that $w(e)\leq t$, where $w(e)$ – Hamming error vector weight, t – code correcting ability.

The generated vector is added to the code word, we get the code word: $c^*=c+e$.

5. A codegram is generated by:

– when shortening characters: adding (extending) characters of the initialization vector: $c_X{}^*=c^*+IV_1$;

– when extending characters: deleting (shortening) characters of the initialization vector: $c_X{}^*=c^*-IV_2$.

6. A flawed text (residue) and a flag (damage) are generated:

$$C_j^* = C_j - C_{k-h_j}, E_{K_{MV2}} : C_j^* \to \left\| f(x)_i \right\| + \left\| C(x)_i \right\|,$$

$$C_j^* = C_{h_r}, E_{K_{MV2}} : C_j^* \to \left\| f(x)_i \right\| + \left\| C(x)_i \right\|.$$

Thus, the presented mathematical apparatus makes it possible to form a complex algorithm for providing basic security services. At the same time, it is proposed to use the approach of implementing the AES GSM algorithm, or ChaCha20 with Poly1305 in the Authenticated Encryption with Associated Data (AEAD) mode. Fig. 2 shows a block diagram of the AEAD mode.

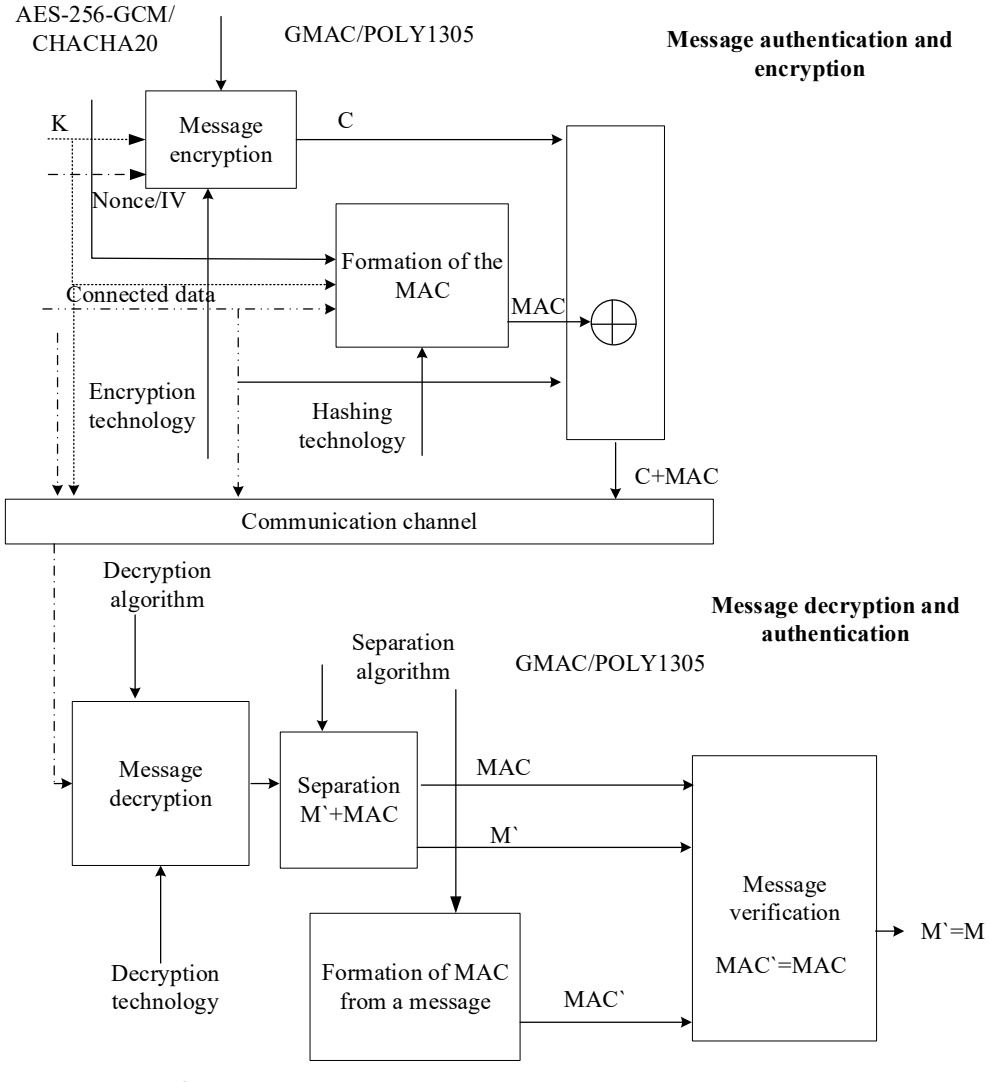


Fig. 2. Block diagram of authenticated encryption with associated data

Thus, the proposed mathematical apparatus of crypto-code constructions and the UMAC cascade hashing algorithm makes it possible to form complex algorithms for providing security services in the SSL/TLS protocol. This ensures the required level of stability in the post-quantum cryptoperiod, efficiency and reliability due to the use of noise-immune codes in the generation of a codegram (cryptogram) in CCC.

5. Development of an improved SSL/TLS protocol based on post-quantum algorithms

5. 1. Development of complex algorithms for providing security services based on post-quantum algorithms

It is proposed to use the synthesis of post-quantum cryptography algorithms with the improved UMAC algorithm as the basis for the formation of a compensated algorithm for providing security services in the SSL/TLS protocol, which will provide the required level of security, efficiency and reliability in the post-quantum crypto period. At the same time, a distinctive feature of the proposed approach is a change in the principle of generating an authenticated message. At the beginning, data is fed to the cascade hashing algorithm, and parameters for crypto-code constructions, initialization vectors of modified elliptic codes are formed in parallel. This approach provides the maximum conversion speed and allows parallelizing the tasks of encryption and generation of the MAC code. On the receiving side, transformations are also performed in parallel mode, which provides the required level of efficiency. Fig. 3 shows a block diagram of the formation of an authenticated message, as well as in the form of an algorithm:

Step 1. Generating a private session key: selecting elliptic curve parameters:

$$K=\{a_1,a_2,\dots a_6\}, a_i \forall GF(q),$$

$$\text{Private key } G^{EC_i}, X^i, P^i, D^i, e. \quad (1)$$

In addition, if necessary, initialization vectors are formed – $|IV_1|$, $|IV_2$ (for parameterization and generation of shortened and/or extended modified elliptic codes).

Inputting information to the first layer of the cascade UMAC algorithm:

$$Hash_{L1_i} = Hash_{L1_i} + {}_{+64}\left(\left(M_{i_{j+0}} {}_{+32} K_{L1_{j+0}}\right) \times_{64} \left(M_{i_{j+4}} {}_{+32} K_{L1_{j+4}}\right)\right),$$

$$Hash_{L1_i} = Hash_{L1_i} + {}_{+64}\left(\left(M_{i_{j+1}} {}_{+32} K_{L1_{j+1}}\right) \times_{64} \left(M_{i_{j+5}} {}_{+32} K_{L1_{j+5}}\right)\right),$$

$$Hash_{L1_i} = Hash_{L1_i} + {}_{+64}\left(\left(M_{i_{j+2}} {}_{+32} K_{L1_{j+2}}\right) \times_{64} \left(M_{i_{j+6}} {}_{+32} K_{L1_{j+6}}\right)\right),$$

$$Hash_{L1_i} = Hash_{L1_i} + {}_{+64}\left(\left(M_{i_{j+3}} {}_{+32} K_{L1_{j+3}}\right) \times_{64} \left(M_{i_{j+7}} {}_{+32} K_{L1_{j+7}}\right)\right). \quad (2)$$

Step 2. Formation of a hash code on the second layer of the cascade hashing algorithm

$$Y_{L2} = \left(M_{P_n} + kM_{P_{n-1}} + \dots + k^{n-1}M_{P_1} + k^n\right) \bmod (p),$$

which is calculated by an iterative procedure (for all i=1, 2, ..., n):

$$Poly_i = \left(kPoly_{i-1} + M_{P_i}\right) \bmod (p),$$

$$Poly_0 = 1, p = prime(Wordbits),$$

using Horner's scheme:

$$M_{P_n} + kM_{P_{n-1}} + \dots + k^{n-1}M_{P_1} + k^n = \left(\left(\left(k + M_{P_1}\right)k + M_{P_2}\right)k + \dots + M_{P_{n-1}}\right)k + M_{P_n}. \quad (3)$$

Step 3. Encryption of the plain text M based on MEC in the McEliece crypto-code construction:

$$c_x^* = M_i \times G_X^{EC_i} + e, \quad (4)$$

where $G_X^{EC_i} = X^i \times G^{EC_i} \times P^i \times D^i$ – public key.

Formation of the MAC code using Pad based on CCC on MEC:

$$Y_{L3} = \left(\left(\left(\sum_{i=1}^{m} Y_{L2_i} K_{L3_{1i}}\right) \bmod \left(prime(36)\right)\right) \bmod \left(2^{32}\right)\right) xor\left(K_{L3_2}\right),$$

$$Y = Y_{L3} \oplus Pad. \quad (5)$$

Step 4. Formation of an authenticated message based on the concatenation algorithm:

$$C_{c_x}^Y = Y \,\|\, c_x^*. \quad (6)$$

When using the proposed approach in smart technologies, the McEliece's (Niederreiter's) hybrid crypto-code construction (HCCC) on flawed codes is used, which makes it possible to reduce energy and computing costs while maintaining stability. To maintain the required level of resistance, the technology of damaging and dividing into two channels is used: transmission of damage, and transmission of flawed text. The block diagram of the formation of an authenticated message based on HCCC on flawed codes is shown in Fig. 4.

The main difference from the algorithm based on modified elliptic codes is the infliction of damage after the formation of an authenticated message. Thus, an additional step is added, at which the MV2 algorithm is used [17, 29, 30], the block diagram is shown in Fig. 5.

The main advantage in the proposed methods and protocols for providing security services based on the use of flawed codes is the use of not block symmetric ciphers (BSC), but McEliece and Niederreiter CCC to ensure the cryptographic strength of damage and/or flawed text.

The uniqueness distance for a random cipher model, for which there is a probability of obtaining a meaningful text with a random and equiprobable choice of the key K and an attempt to decrypt the ciphertext, with

$$N_S = H(K)\frac{2^{HL}}{|I|^L} = 1, L = U_0 = \\ = \frac{H(K)}{\log|I| - H} = \frac{H(K)}{B\log|I|}, \quad (7)$$

where **B** – source text redundancy; **H** – entropy per letter of a meaningful text in the input alphabet I, $|I|>2$, 2^{HL} – approximate value of the number of meaningful texts.

In [29, 30], a cyclic algorithm for obtaining flawed texts means a universal mechanism for causing damage (C_m, where m – number of cycles). The algorithm consists in randomly replacing the bit representation of each character of the source text with a tuple of a smaller or equal number of bits, followed by their concatenation. Fig. 6 shows a universal mechanism for causing damage (*MV2* algorithm (flawed text generation)).

The presented algorithms allow implementing and generating an authenticated message based on post-quantum algorithms. This provides not only the required level of stability and speed (efficiency) of information processing, but also increases the level of reliability using noise-resistant algebrogeometric codes. Thus, the necessary enhancements to the SSL/TLS protocol are provided.

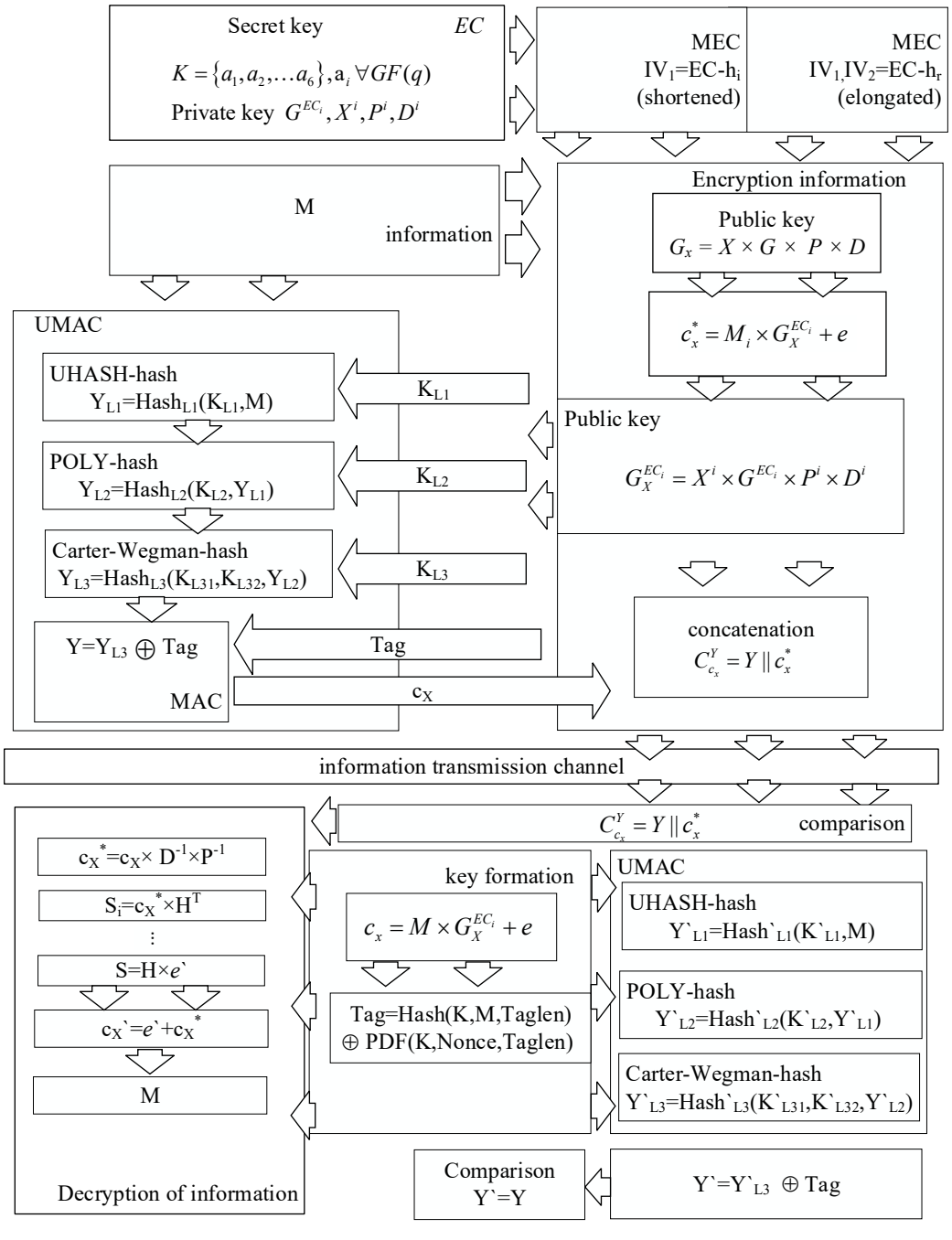


Fig. 3. Block diagram of the formation of an authenticated message based on crypto-code constructions on modified elliptic codes

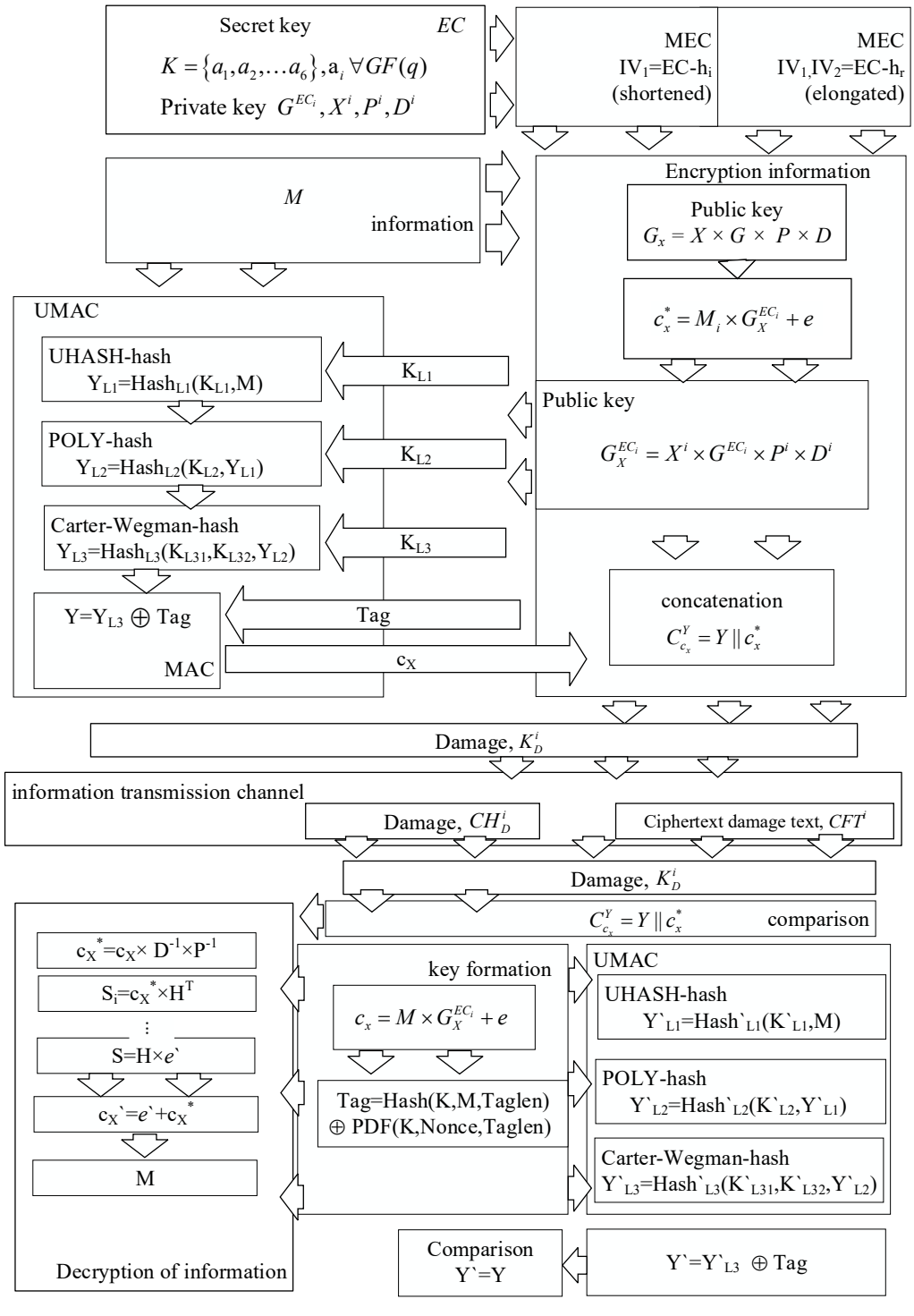


Fig. 4. Block diagram of the formation of an authenticated message based on hybrid crypto-code constructions on flawed codes

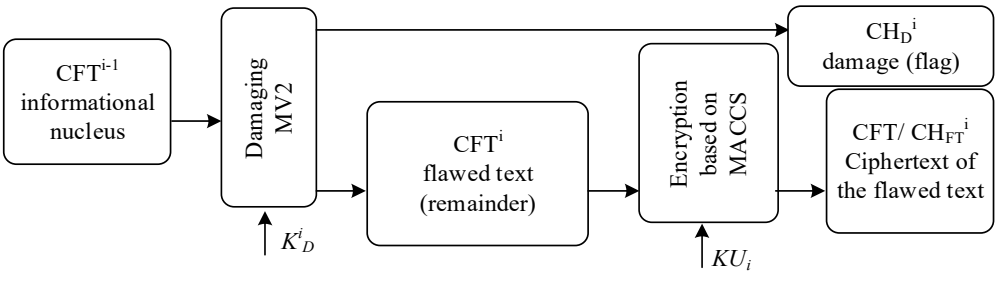
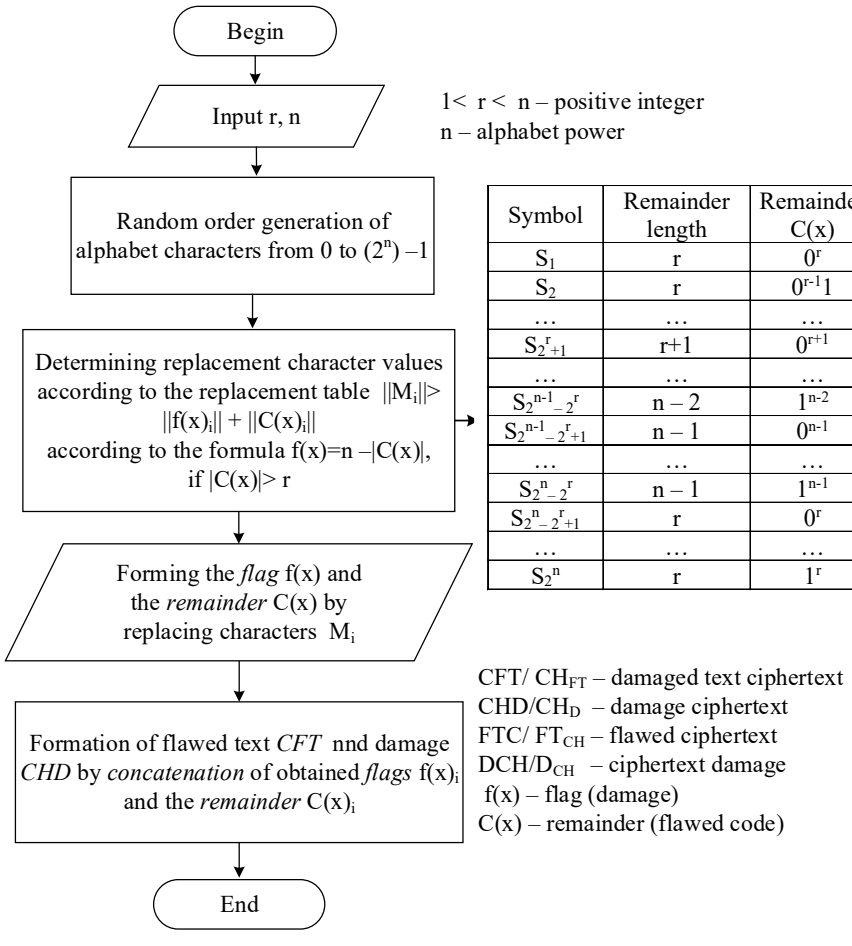


Fig. 5. Block diagram of the MV2 algorithm



Symbol	Remainder length	Remainder C(x)	Flag f(x)
S_1	r	0^r	$0^{n-r-1}1$
S_2	r	$0^{r-1}1$	$0^{n-r-1}1$
…	…	…	…
S_{2^r+1}	r+1	0^{r+1}	$0^{n-r-2}1$
…	…	…	…
$S_{2^{n-1}-2^r}$	n – 2	1^{n-2}	01
$S_{2^{n-1}-2^r+1}$	n – 1	0^{n-1}	1
…	…	…	…
$S_{2^n-2^r}$	n – 1	1^{n-1}	1
$S_{2^n-2^r+1}$	r	0^r	0^{n-r}
…	…	…	…
S_{2^n}	r	1^r	0^{n-r}

Fig. 6. The *MV2* algorithm

5. 2. Justification of the direction of improvement of the block diagram of the SSL/TLS protocol

At the heart of the SSL/TLS protocol enhancements are complex algorithms based on post-quantum algorithms, which allow the provision of security services, taking into account the elimination of vulnerabilities based on the use of the handshake phase.

At the present stage, two versions of the SSL/TLS protocol v.1.2, v.1.3 are used, their architecture consists of two protocols:

I – handshake protocol (purpose – authentication and key exchange), under which the Client and the Server perform the following procedures:

– protocol version negotiation;

– selection of a cryptographic algorithm or cipher suite;

– authentication with asymmetric cryptography;

– determining the shared secret key to be used for symmetric encryption at the next level.

II – recording protocol. At this level, the following procedures are performed:

– all outgoing messages are encrypted with the secret key set during the handshake;

– the encrypted messages are transmitted from the Client to the Server;

– the server checks the received encrypted messages for changes;

– if no changes are made, the encrypted messages are decrypted using the secret key.

To ensure that the encrypted message has not been modified during transmission, the TLS v.1.3 protocols use authenticated encryption (Authenticated Encryption with Associated Data mode, AEAD), Fig. 7, 8 show the differences between the versions of the SSL/TLS protocol stack.

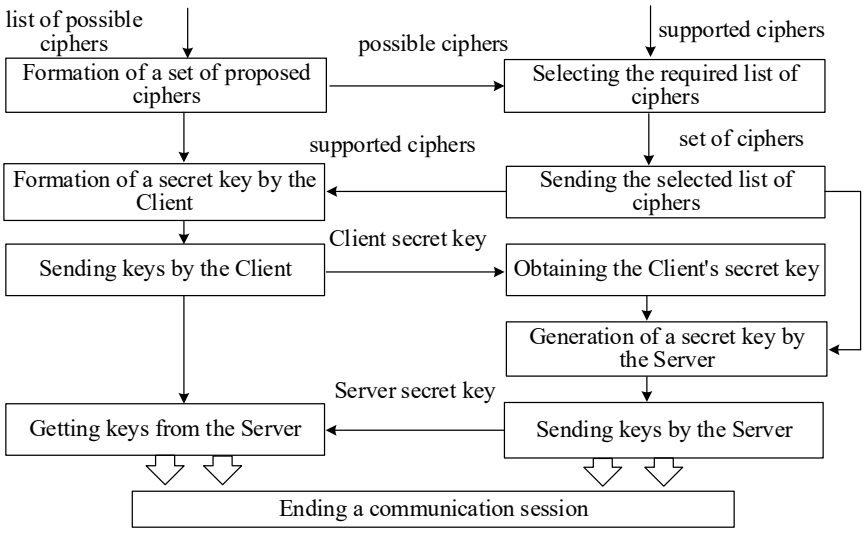


Fig. 7. Block diagram of the SSL/TLS protocol, version 1.2

The fundamental difference between version 1.2 and version 1.3 is the rejection of the choice of various symmetric encryption algorithms (version 1.2 was the choice) and the use of complex algorithms for generating an authenticated message in AEAD mode (version 1.3).

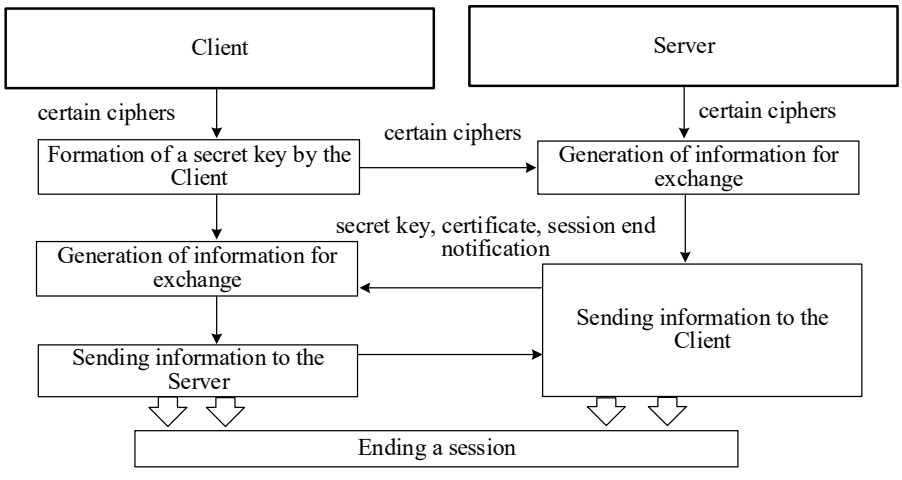


Fig. 8. Block diagram of the SSL/TLS protocol, version 1.3

However, even in version 1.3, the handshake protocol (phase) remains vulnerable, in which a secret (shared) key is exchanged for general use in the selected symmetric cryptoalgorithm, both on the server and on the client side.

In addition, the algorithms used in version 1.2 and version 1.3 are not post-quantum algorithms and cannot guarantee the required level of security in the context of the emergence of a full-scale quantum computer. To provide security services and eliminate vulnerabilities in the handshake phase of the SSL/TLS protocol, it is proposed to transmit only the parameters of the curve equation, as well as, if necessary, initialization vectors of modified elliptic codes.

Thus, to generate key data on both the server and client sides, it is enough to transfer only the coefficients a_1–a_6, that is, a binary sequence of five characters. At the same time, both sides will be able to form the necessary matrices and be ready to exchange information. Also, there is no need to use additional asymmetric algorithms (Diffie-Hellman, RSA) to transfer key data for a symmetric algorithm (in the proposed case, asymmetric cryptosystems are used with a crypto transformation rate comparable to symmetric cryptoalgorithms). Fig. 9 shows an improved scheme of the SSL/TLS protocol.

If it is necessary to "restore" the session in 0-RTT mode, there is also no need to exchange key data and part of the encrypted code to verify that the shared key is correctly determined. It is enough for the client to send only the curve coefficients for the server to determine the key data for CCC. The use of the error vector e when transmitting information allows it to be considered as a session key for each individual packet, which significantly increases the security level. Fig. 10 shows a block diagram of the enhanced SSL/TLS protocol in 0-RTT mode.

Thus, an improvement is proposed for the most common protocol for providing security services at the transport level SSL/TLS based on post-quantum algorithms – crypto-code constructions based on modified (shortened and/or extended) elliptic codes. This approach significantly reduces the "possibility" of known vulnerabilities to the SSL/TLS protocol. This ensures the required level of security in the post-quantum cryptoperiod, computational and energy-intensive requirements for use in cyber-physical systems based on smart technologies.

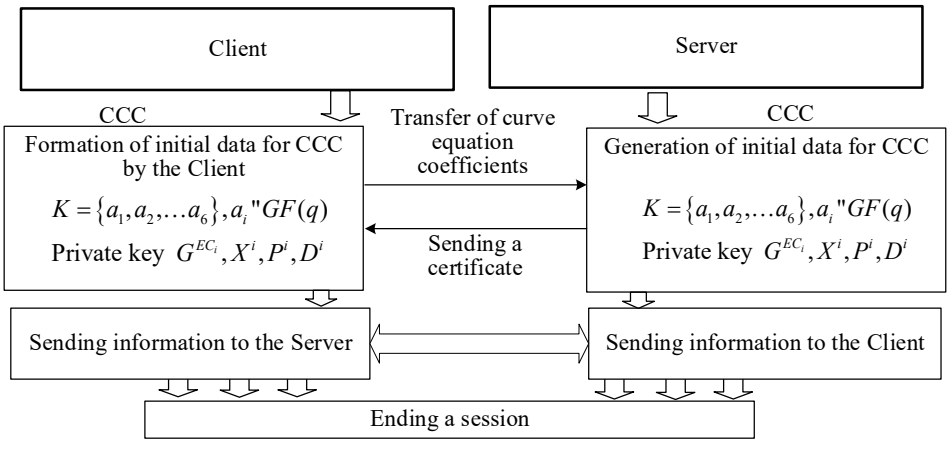


Fig. 9. Improved SSL/TLS protocol scheme

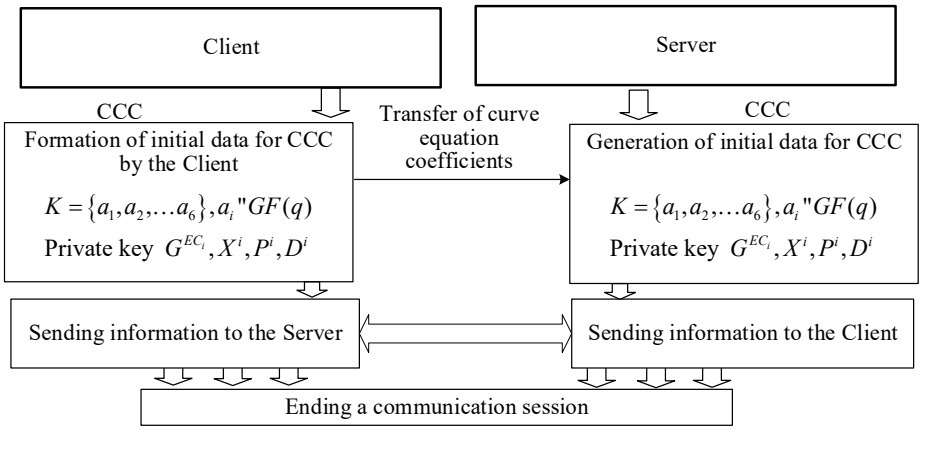


Fig. 10. Block diagram of the enhanced SSL/TLS protocol in 0-RTT mode

Solving problems related to the implementation of the presented protocols, including modules with different requirements for speed, real-time operation, complex processing of information related to cryptographic transformations, etc., cannot rely only on the increasing power of modern computers. Modern software tools and systems designed to solve this range of tasks have a number of rather stringent requirements for resource efficiency, while their quality characteristics, such as time efficiency and resource intensity, are among the determining ones.

The effectiveness of the proposed algorithmic solutions largely depends on the characteristics of the software systems chosen for implementation.

An analysis of the requirements in terms of implementing various modules that determine the effectiveness of the development and operation of the proposed algorithms led to the choice of C# as the main programming language – simple, modern, object-oriented, providing type safety. Being an object-oriented language, C# also provides support for component-oriented programming. The software structure of the proposed solutions is largely based on independent modules. The essence of these components is that they are part of a general model based on properties, methods and events. C# provides language constructs that directly support these concepts, making it a natural language for creating and applying software components in the field under consideration.

5. 3. Estimation of energy costs and resistance of the software implementation of the proposed hybrid cryptosystem

To evaluate time and speed indicators, it is customary to use the unit of measurement *cpb*, where *cpb* (*cycles per byte*) – the number of processor cycles that must be spent to process 1 byte of incoming information.

The complexity of the algorithm is calculated by the expression:

$$Per = Utl \times CPU\_clock / Rate, \quad (8)$$

where *Utl* – processor core utilization (%); *Rate* – algorithm throughput (bytes/s).

Table 3 shows the results of studying the dependence of the length of the code sequence of the algebrogeometric code in the McEliece CCC on the number of processor cycles to perform elementary operations in the software implementation of crypto-code systems.

Table 4 shows the results of studying the dependence of the length of the input sequence on the *MV2* algorithm on the number of processor cycles to perform elementary operations in the software implementation.

Table 3

Results of studies of the dependence of the code sequence length in the McEliece CCC on the number of processor cycles

Length of the code sequence		McEliece on shortened codes			McEliece on extended codes		
		10	100	1,000	10	100	1,000
Number of function calls implementing elementary operations	Character reading	10,294,397	28,750,457	76,759,874	11,432,131	33,460,317	80,859,933
	String comparison	3,406, 921	9,246,748	25,478,498	3,673,756	12,119,867	26,364,634
	String concatenation	1,705,544	5,045,748	12,379,422	1,947,681	6,114,478	13,415,329
Sum		15,406, 862	43,042,953	114,617,794	16,516,381	17,053,568	51,694, 662
Duration of function execution* in processor cycles	Character reading	295,374	810,478	2,001,167	300,479	843,705	2,745,148
	String comparison	178,814	531,379	1,248,684	213,478	561, 754	1,739,170
	String concatenation	544,990	1,328,114	3,586,486	578,174	1,647,638	4,007,883
Sum		1,006,781	2,749,548	7,247,488	1,040,298	109,157	1,092,131
Execution time** in ms		0.52	1.37	3.4	0.55	0.56	1.55

Table 4

Results of studies of the dependence of the input sequence length on the *MV2* algorithm on the number of processor cycles

Length of the code sequence		*MV2*		
		10	100	1,000
Number of function calls implementing elementary operations	Summation	3,942	28,673	275,499
	Subtraction	1,794	3,810	23,881
	Division	3,274	4,804	20,104
	Multiplication	19	109	1,009
	Comparison	8,939	60,963	578,784
Sum		17,968	98,359	899,277
Duration of function execution* in milliseconds	Summation	19.53	93.58	2297.36
	Subtraction	8.89	12.43	199.14
	Division	16.22	15.68	167.65
	Multiplication	0.09	0.36	8.41
	Comparison	44.28	198.96	4826.43
Sum		89	321	7,499
Execution duration** in milliseconds		89	321	7,499

*Note: * – duration of 1,000 operations in processor cycles: character reading – 27 cycles, string comparison – 54 cycles, string concatenation – 297 cycles; ** – a processor with a clock frequency of 2 GHz was taken for the calculation, taking into account the operating system load of 5* **%**

Table 5 demonstrates the results of studies on evaluating the time and speed indicators of information generation and decoding procedures in asymmetric crypto-code systems based on McEliece CCC.

Table 5

Results of studies on the evaluation of time and speed indicators of information generation and decoding procedures

Cryptosystem	Length of the code sequence	Algorithm throughput, Rate (byte/sec)	Processor core utilization (%)	Algorithm complexity, Per (cpb)
McEliece CCC on *EC*	100	46,125,790	56	61.5
	1,000	120,639,896	56	62.0
McEliece CCC on extended *MEC*	100	51,694,662	56	61.7
	1,000	126,399,560	56	62.2
McEliece CCC on shortened *MEC*	100	46,125,790	56	61.5
	1,000	120,639,896	56	62.0

Analysis of Tables 3–5 allows us to conclude that CCC on MEC provides a reduction in energy costs and volumes of key user data. The formation of hybrid cryptosystems based on McEliece CCC on flawed codes can significantly reduce the power of the alphabet ($GF(2^4–2^6)$) without reducing the level of cryptographic strength due to the use of multichannel cryptography systems on flawed codes. Procedures for using the MV2 damage algorithm have virtually no effect on the encryption speed in McElice CCC.

To conduct statistical research on the stability of the studied cryptosystems, the NIST STS 822 package is used [17]. The research results are presented in Table 6.

The indicators given in Table 6 showed that, despite the decrease in the Galois field power to $GF(2^6)$ for CCC and $GF(2^4)$ for HCCC, respectively, the statistical characteristics of such crypto-code structures turned out to be at least no worse than the traditional McEliece CCC on $GF(2^{10})$. All cryptosystems passed 100 % of the tests, and the HCCC on shortened MEC showed the best result: 155 out of 189 tests were passed at the level of 0.99, which is 82 % of the total number of tests. At the same time, the traditional McEliece CCC on $GF(2^{10})$ showed 149 tests at the level of 0.99.

6. Discussion of the results of stability and performance of the complexed algorithm

The post-quantum cryptoalgorithms based on crypto-code constructions on modified (flawed) codes proposed in the paper make it possible to obtain the maximum number of emergent properties with minimal resource costs aimed at generating a synergistic security effect in the system. The stability and energy intensity of the proposed crypto-code constructions (Tables 3–6) provide the possibility of their practical use in cyber-physical systems based on the integration of smart technologies, the Internet of Things and wireless channels. The proposed solutions for providing security services based on complex asymmetric cryptosystems with a cascade hashing algorithm based on crypto-code constructions on modified elliptic codes (Fig. 3), flawed codes (Fig. 4) provide practical use in the SSL/TLS protocol.

To simulate crypto-code constructions, elliptic codes over the Galois field $GF(2^4)$ were used, which makes it possible to provide the required level of security and implement the proposed approach to the formation of an improved SSL/TLS protocol, as well as to ensure the formation of a cascade UMAC hashing algorithm. For modeling, the C# programming language was used, which makes it possible to form a library of the mathematical apparatus of coding theory over the Galois field $GF(2^4)$. In addition, the software package allows you to study the stability and performance of the proposed approach – post-quantum algorithms on EC (MEC) codes.

The main difference between the proposed hybrid crypto-code constructions (HCCC) and the "classical" hybrid cryptosystem is the use of an asymmetric cryptosystem (provable security model) based on the McEliece CCC as the main encryption mechanism, rather than the BSC (temporal security model).

Thus, provable security is provided at an encryption rate comparable to cryptotransformations in the BSC and reliability is integrated by using error-correcting codes on elliptic curves. To reduce energy consumption, the *MV2* algorithm is used, which provides increases the entropy of the ciphertext and allows a message to be transmitted over one channel (the damage vector can be used in the McEliece CCC as the error vector e), or over two independent channels. Thus, the use of the *MV2* algorithm increases the cryptographic strength of the system, allows you to "reduce" the cardinality of the alphabet (the dimension of the $GF(2^m)$ field for constructing McEliece MCCC) without reducing the cryptographic strength of the system as a whole.

So, the proposed approach makes it possible to encrypt significant amounts of data circulating in information and communication systems without reducing the levels of security and performance, as well as to ensure the required level of reliability.

7. Conclusions

1. Complex algorithms based on post-quantum algorithms – McEliece (Niederreiter) crypto-code constructions on MEC (flawed) codes with an improved UMAC algorithm are proposed. Such synthesis provides confidentiality, integrity and authenticity services. The studies of post-quantum algorithms provide the required level of security, which is confirmed by the results of testing based on the NIST STS 822 package. In addition, crypto-code constructions provide performance indicators at the level of symmetric encryption algorithms, which allows for the required level of performance of crypto transformations and practical implementation for use in cyber-physical systems. This approach allows using them not only in this protocol, but also in smart technologies with limited computing resources. Also, based on the use of noise-resistant algebro-geometric codes, an integrated increase in the reliability of transmitted data is provided.

2. Directions for improving the block diagram of the SSL/TLS protocol based on complex algorithms – McEliece (Niederreiter) crypto-code constructions on MEC (flawed codes) with an improved cascade hashing algorithm are proposed. This approach significantly reduces the "possibilities" of known vulnerabilities to the SSL/TLS protocol by using only asymmetric cryptosystems, and "simplifying" the handshake phase. The proposed protocol improvement eliminates the need to exchange a separate key before data transmission

and use asymmetric encryption algorithms to exchange key data (certificates). This ensures the required level of security in the post-quantum cryptoperiod, computational and energy-intensity requirements for use in cyber-physical systems based on smart technologies.

3. The studies on assessing the energy intensity of the costs of software implementation of the proposed hybrid cryptosystem provide practical implementation over the $GF(2^4)$ field, which allows using them in cyber-physical systems based on smart technologies with limited computing resources. This ensures the required level of resistance, which is confirmed by the results of testing based on the NIST STS-822 package. All cryptosystems passed 100 % of the tests, while the best result was shown by the SCCC on shortened MEC: 155 out of 189 tests were passed at the level of 0.99, which is 82 % of the total number of tests.

Conflict of interest

The authors declare that they have no conflict of interest regarding this study, whether financial, personal, authorship or otherwise that could affect the research and its results presented in this paper.

Financing

The study was conducted without financial support.

Data availability

The manuscript has no associated data.

References

1. Arora, J. et al. (2023). Securing web documents by using piggybacked framework based on Newton's forward interpolation method. Journal of Information Security and Applications, 75, 103498. doi: https://doi.org/10.1016/j.jisa.2023.103498
2. Yevseiev, S., Hryshchuk, R., Molodetska, K., Nazarkevych, M., Hrytsyk, V., Milov, O. et al.; Yevseiev, S., Hryshchuk, R., Molodetska, K., Nazarkevych, M. (Eds.) (2022). Modeling of security systems for critical infrastructure facilities. Kharkiv: PC TECHNOLOGY CENTER, 196. doi: https://doi.org/10.15587/978-617-7319-57-2
3. Saribas, S., Tonyali, S. (2022). Performance Evaluation of TLS 1.3 Handshake on Resource-Constrained Devices Using NIST's Third Round Post-Quantum Key Encapsulation Mechanisms and Digital Signatures. 2022 7th International Conference on Computer Science and Engineering (UBMK). doi: https://doi.org/10.1109/ubmk55850.2022.9919545
4. Khan, N. A., Khan, A. S., Kar, H. A., Ahmad, Z., Tarmizi, S., Julaihi, A. A. (2022). Employing Public Key Infrastructure to Encapsulate Messages During Transport Layer Security Handshake Procedure. 2022 Applied Informatics International Conference (AiIC). doi: https://doi.org/10.1109/aiic54368.2022.9914605
5. Ramraj, S., Usha, G. (2023). Signature identification and user activity analysis on WhatsApp Web through network data. Microprocessors and Microsystems, 97, 104756. doi: https://doi.org/10.1016/j.micpro.2023.104756
6. Nie, P., Wan, C., Zhu, J., Lin, Z., Chen, Y., Su, Z. (2023). Coverage-directed Differential Testing of X.509 Certificate Validation in SSL/TLS Implementations. ACM Transactions on Software Engineering and Methodology, 32 (1), 1–32. doi: https://doi.org/10.1145/3510416
7. Berbecaru, D. G., Petraglia, G. (2023). TLS-Monitor: A Monitor for TLS Attacks. 2023 IEEE 20th Consumer Communications & Networking Conference (CCNC). https://doi.org/10.1109/ccnc51644.2023.10059989
8. Wang, K., Zheng, Y., Zhang, Q., Bai, G., Qin, M., Zhang, D., Dong, J. S. (2022). Assessing certificate validation user interfaces of WPA supplicants. Proceedings of the 28th Annual International Conference on Mobile Computing And Networking. doi: https://doi.org/10.1145/3495243.3517026
9. Kottur, S. Z., Kadiyala, K., Tammana, P., Shah, R. (2022). Implementing ChaCha based crypto primitives on programmable SmartNICs. Proceedings of the ACM SIGCOMM Workshop on Formal Foundations and Security of Programmable Network Infrastructures. doi: https://doi.org/10.1145/3528082.3544833
10. Chen, L., Li, X., Yang, Z., Qian, S. (2022). Blockchain-based high transparent PKI authentication protocol. Chinese Journal of Network and Information Security, 8 (4), 1–11. doi: https://doi.org/10.11959/j.issn.2096-109x.2022052
11. Zhang, Z., Zhang, H., Wang, J., Hu, X., Li, J., Yu, W. et al. (2023). QKPT: Securing Your Private Keys in Cloud With Performance, Scalability and Transparency. IEEE Transactions on Dependable and Secure Computing, 20 (1), 478–491. doi: https://doi.org/10.1109/tdsc.2021.3137403
12. Zhou, Z., Bin, H., Li, J., Yin, Y., Chen, X., Ma, J., Yao, L. (2022). Malicious encrypted traffic features extraction model based on unsupervised feature adaptive learning. Journal of Computer Virology and Hacking Techniques, 18 (4), 453–463. doi: https://doi.org/10.1007/s11416-022-00429-y
13. Bertok, C., Huszti, A., Kovacs, S., Olah, N. (2022). Provably secure identity-based remote password registration. Publicationes Mathematicae Debrecen, 100, 533–565. doi: https://doi.org/10.5486/pmd.2022.suppl.1
14. Aayush, A., Aryan, Y., Muniyal, B. (2022). Understanding SSL Protocol and Its Cryptographic Weaknesses. 2022 3rd International Conference on Intelligent Engineering and Management (ICIEM). doi: https://doi.org/10.1109/iciem54221.2022.9853153
15. Guo, S., Zhang, F., Song, Z., Zhao, Z., Zhao, X., Wang, X., Luo, X. (2022). Detection of SSL/TLS protocol attacks based on flow spectrum theory. Chinese Journal of Network and Information Security, 8 (1), 30–40. doi: https://doi.org/10.11959/j.issn.2096-109x.2022004
16. Arunkumar, B., Kousalya, G. (2022). Secure and Light Weight Elliptic Curve Cipher Suites in SSL/TLS. Computer Systems Science and Engineering, 40 (1), 179–190. doi: https://doi.org/10.32604/csse.2022.018166

17. Yevseiev, S., Ponomarenko, V., Laptiev, O., Milov, O., Korol, O., Milevskyi, S. et al.; Yevseiev, S., Ponomarenko, V., Laptiev, O., Milov, O. (Eds.) (2021). Synergy of building cybersecurity systems. Kharkiv: PC TECHNOLOGY CENTER, 188. doi: https://doi.org/10.15587/978-617-7319-31-2
18. Gavrilova, A., Volkov, I., Kozhedub, Y., Korolev, R., Lezik, O., Medvediev, V. et al. (2020). Development of a modified UMAC algorithm based on crypto¬code constructions. Eastern-European Journal of Enterprise Technologies, 4 (9 (106)), 45–63. doi: https://doi.org/10.15587/1729-4061.2020.210683
19. Guide for Cybersecurity Event Recovery. NIST. Available at: https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-184.pdf
20. Security requirements for cryptographic modules. Available at: https://csrc.nist.gov/csrc/media/publications/fips/140/2/final/documents/fips1402.pdf
21. Guide to LTE Security. Available at: https://csrc.nist.gov/csrc/media/publications/sp/800-187/draft/documents/sp800_187_draft.pdf
22. Report on Post-Quantum Cryptography. Available at: https://csrc.nist.gov/publications/detail/nistir/8105/final
23. Bernstein, D. J., Buchmann, J., Dahmen, E. (Eds.). (2009). Post-Quantum Cryptography. Springer. doi: https://doi.org/10.1007/978-3-540-88702-7
24. Pohasii, S., Yevseiev, S., Zhuchenko, O., Milov, O., Lysechko, V., Kovalenko, O. et al. (2022). Development of crypto-code constructs based on LDPC codes. Eastern-European Journal of Enterprise Technologies, 2 (9 (116)), 44–59. doi: https://doi.org/10.15587/1729-4061.2022.254545
25. Korol, O., Havrylova, A., Yevseiev, S. (2019). Practical UMAC algorithms based on crypto code designs. Przetwarzanie, transmisja i bezpieczenstwo informacji. Vol. 2. Bielsko-Biala: Wydawnictwo naukowe Akademii Techniczno-Humanistycznej w Bielsku-Bialej, 221–232.
26. Carter, J. L., Wegman, M. N. (1979). Universal classes of hash functions. Journal of Computer and System Sciences, 18 (2), 143–154. doi: https://doi.org/10.1016/0022-0000(79)90044-8
27. Bierbrauer, J., Johansson, T., Kabatianskii, G., Smeets, B. (2001). On Families of Hash Functions via Geometric Codes and Concatenation. Lecture Notes in Computer Science, 331–342. doi: https://doi.org/10.1007/3-540-48329-2_28
28. Bettaieb, S., Bidoux, L., Blazy, O., Cottier, B., Pointcheval, D. (2023). Post-quantum and UC-Secure Oblivious Transfer from SPHF with Grey Zone. Lecture Notes in Computer Science, 54–70. doi: https://doi.org/10.1007/978-3-031-30122-3_4
29. Mishhenko, V. A., Vilanskij, Ju. V., (2007). Ushherbnye teksty i mnogokanal'naja kriptografija [Damaged texts and multichannel cryptography]. Minsk: Jenciklopediks, 292.
30. Mishhenko, V. A., Vilanskij, Ju. V., Lepin, V. V. (2006) “Kriptograficheskij algoritm MV 2 [Cryptographic algorithm MV 2]. Minsk: Jenciklopediks, 176.