КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (22), 2023
ISSN 2663 - 4023
DOI 10.28925/2663-4023.2023.22.122133
UDC 004.056.5:510.22(043.3)

Anna Ilyenko
PhD, assistant professor, assistant professor of Information Security Systems Department National Aviation University of Kyiv, Faculty of Cyber Security and Software Engineering, Kyiv, Ukraine
ORCID 0000-0001-8565-1117
ilyenko.a.v@nau.edu.ua

Sergii Ilyenko
PhD, assistant professor, assistant professor of Automation and Energy Management Department National Aviation University of Kyiv, Aerospace Faculty, Kyiv, Ukraine
ORCID 0000-0002-0437-0995
ilyenko.s.s@nau.edu.ua

Olena Prokopenko
Senior teacher of the Department of Computerized Information Protection Systems
National Aviation University, Kyiv, Ukraine
ORCID 0000-0001-9895-888X
bortnik.olena.v@nau.edu.ua

Iryna Kravchuk
Assistant of the Department of Computerized Information Protection Systems
National Aviation University, Kyiv, Ukraine
iryna.kravchuk@npp.nau.edu.ua

PRACTICAL APPROACHES TO ORGANIZING SECURE DATA TRANSFER VIA TLS PROTOCOL USING OPENSSL MEANS

Annotation. Information protection during message transmission is one of the most important tasks in the modern world. The workstations of a particular organization can be well protected using different software and hardware means, but when information is transferred to an open unprotected space, there is a high probability of data leakage, interception, and substitution. In most cases, the use of insufficiently effective security measures causes the loss of personal data of citizens, data of an enterprise or organization that is a commercial secret, information with limited access or even state secrets of the critical infrastructure sector. In this case, it is considered appropriate to use modern cryptographic methods to organize secure data transmission. Considering the ever-growing statistics of cyberattacks on information and telecommunication networks, after in-depth analysis and development of this issue, the authors of the article highlighted the current directions of protection of information and telecommunication networks and security solutions in information and telecommunication networks. The authors have comprehensively covered and investigated the basic principles of the modern state of data transmission security and the organization of information protection during its transmission using the TLS protocol, which made it possible to determine the directions for solving or modernizing existing information security means. Also, attention is paid to the development of a software implementation of the module for secure data transmission in the information and telecommunications network using the improved TLS protocol by means of OpenSSL, which made it possible to establish connections using digital signature algorithms. The authors are planning a series of scientific and technical solutions for the development and implementation of effective cryptographic methods to ensure the security of information and telecommunications networks.

Keywords: network; encryption; protocol; digital signature; data protection.

ISSN 2663 - 4023

INTRODUCTION

At the current stage of society's development, information protection is one of the most important tasks of the present day. Along with the rapid development of society's informatization, the level of vulnerability of the systems in which information is stored is also growing. This trend in the development of the information sphere encourages the development of cybercrime at various levels, ranging from online fraud to cybercrime at the state level and critical infrastructure facilities where information of increased importance circulates.

One of the options to solve this problem is the use of cryptographic methods of information protection as one of the components of the complex of security tools at various stages of the functioning of information networks and during the transmission, storage, and processing of information. It has been theoretically proven that cryptosystems allow to protect data transmission channels and information that circulates and is stored there with a high degree of reliability.

The main means of protecting information by cryptographic methods include software, hardware and software-hardware tools that implement protection algorithms to ensure the following: ensuring the confidentiality, integrity and availability of information circulating in the exchange environment; the decryption procedure for attackers should be as complicated as possible; the content of the transmitted information should not affect the effectiveness of the cryptographic algorithm; the reliability of the protection algorithm should not depend on the content of the secret of the encryption algorithm itself.

Problem statement. After analyzing the existing solutions and the current state of the organization of protected data transmission using cryptographic methods [1], [6]–[8], it can be concluded that for today there are several cryptographic protocols of the TLS/SSL family for organizing remote information exchange. The analysis of recent studies and publications allowed to form the requirements for the improved software module for the organization of protected data transmission using the improved TLS 1.3 protocol by means of OpenSSL. Considering the analysis of recent research and publications on network security and the role of the use of cryptographic tools [9]–[12], relevant is the issue of developing and implementing an improved author's module for ensuring basic criteria for information protection in the implementation of remote data exchange that will meet the requirements of the present. The aim of the work is development and practical implementation of a cryptographic module for secure data transmission using the improved TLS protocol by means of OpenSSL. In the course of the work, a study of modern methods of cryptographic information protection will be carried out; a study of the main characteristics and approaches of the OpenSSL cryptographic package for symmetric and asymmetric encryption; a procedure for improving the TLS1.3 protocol using DSTU 4145-2002 for performing secure data transmission will be developed; an evaluation of the feasibility of implementing a software implementation of the cryptographic module for protected data transmission using the improved TLS protocol by means of OpenSSL will be carried out.

THEORETICAL BASIS OF THE STUDY

To avoid data interception, secure data transfer protocols are widely used today, namely the HTTP protocol, which for security purposes works through SSL or TLS certificates. Its use not only allows to send information from one's own resource with a higher degree of security and ensure reliable protection of the information transferred to this resource, but also reduce the risk of network attacks on external information resources of the server.

The presence of an SSL certificate is becoming one of the most used technical means of protecting information, the original type of site passport, which guarantees the authenticity of

the domain address. The use of this protocol allows to create an information channel protected from substitution, which allows to prevent unauthorized access and provide basic criteria for the security of information and information object [2], [3].

In today's Internet, the tasks of maintaining the confidentiality of important information are entrusted to TLS/SSL. There are currently four versions of TLS, all of which are described in the RFC: TLS 1.0, TLS 1.1, TLS 1.2, and TLS 1.3, which was released in 2018.

In general, the correct operation of the TLS protocol is performed in 2 phases: Handshake and Record. During the Handshake phase, the client and server will: agree on the protocol version; select a cryptographic algorithm or cipher suite; authenticate each other using asymmetric cryptography; determine a common secret key that will be used for symmetric encryption in the next phase. During the Record phase: all outgoing messages will be encrypted using the shared secret key established during the Handshake phase. Then the encrypted messages are transferred to the other party. They are checked to see if any changes occurred during the transfer or not. If not, the messages will be decrypted using the same symmetric secret key [4], [5].

The conducted studies of the structure, implementation features and types of cryptographic algorithms allowed to determine the direction of improvement of the TLS 1.3 protocol using DSTU 4145-2002 to perform secure data transmission. Next, let's review the theoretical description of the solution for creating a software module for protected data transmission, the procedure for initial connection, the main features of the X.509 certificate and the mathematical model for implementing DSTU 4145-2002. During the theoretical comparison of RSA, ECDSA and DSTU 4145 algorithms, their advantages and disadvantages were identified. The RSA algorithm is based on the complexity of the factoring challenge, while ECDSA and DSTU 4145-2002 are based on the use of elliptic curves. The main advantages of DSTU 4145-2002 can be considered a reliable and relatively small size of the public key, which leads to its faster transfer, the verification process is also reduced, which significantly affects the overall performance of the algorithm in the system [12]–[14].

The theoretical implementation of the DSTU 4145-2002 algorithm in the cryptographic module of protected data transfer using the improved TLS protocol assumes the implementation of a qualified digital signature based on the properties of groups of elliptic curves points over the $GF(2m)$ fields. This standard will be used to generate and verify the signature when creating a self-signed X.509 certificate and its child certificate.

An important element of the standard is the choice of general parameters of a digital signature, the parameters can be the same for an arbitrary number of users [15]–[17]. General parameters include:

1. parameters of the main field: degree of expansion m, type of basis (polynomial, optimal normal), if the basis is polynomial, then a primitive polynomial $f(t)$ (trinomial, pentanomial), which determines the polynomial basis;
2. elliptical curve of type $y^3 + xy = x^3 + Ax^2 + B$;
3. base point of the elliptic curve P;
4. identifier of the used hashing function iH (optional parameter);
5. length of the qualified digital signature LD (length of the data block containing the qualified digital signature);
6. order of the base point n.

When selecting the general parameters, one should pay attention to the type of the basis that will be used in the software module. The standard itself provides recommended elliptic curves in the optimal normal basis and recommended elliptic curves in the polynomial basis. Comparing the multiplication of elements in the polynomial basis field, where all actions are performed on words, when multiplying in the normal basis, actions are performed on separate

bits of the operands, moreover, on different bits of two operands. Therefore, the software implementation of multiplication in the normal basis will be slower than the software implementation in the polynomial basis [12], [13].

Based on the analysis of the time costs for the multiplication operation in the polynomial and normal basis, it can be concluded that the use of the polynomial basis is more appropriate, and it should also be noted that if a field in which it is necessary to spend fewer cycles to calculate the inverse elements than in at least one field with a lower sequence number (with a lower m) is considered better, then the field with polynomial powers m=257 is the best to use.

The calculation of qualified digital signature keys is divided into two stages: finding the public and private keys. The conditions for the calculation and storage of a private digital signature key shall make it impossible to provide unauthorized access to the private key or a part of it, as well as to intermediate data used in the process of calculating the private key. The conditions for storing a private key should make it impossible to modify, destroy or replace it.

Proceeding from the task of the work and analysis of the current state of organization of a secure connection, a practical cryptographic module for secure data transmission using the improved TLS protocol by means of OpenSSL is proposed. The client-server windowed application with an upgraded module for data transmission over the secure TLS 1.3 protocol implements the following features:

- For TLS, it is provided to use RSA (RSA-PSS-RSAE-SHA256), ECDSA (ECDSA-SECP256r1-SHA256) signatures (available by default), as well as to implement the signature algorithm of DSTU 4145-2002 (curve in the polynomial basis of 257 bits);
- for each algorithm, the outdated GOST 34.311 hash (based on MD4) was replaced by the SHA256 hashing algorithm;
- for each algorithm, the creation of an X.509 CA certificate (RSA - 3072 bits, ECDSA - NIST P-256r1, DSTU - 4145-2002 M257-PB) is provides and a child certificate using the same algorithm for the server part;
- the application implements an interface where from the server or client side it is possible to select the desired matching algorithm for a given session, and from the client side the address of the server that will accept the connection is indicated;
- the application waits for data input from the client, after which it transmits this message via an encrypted channel, which, after decryption, is displayed on the server.

EXPERIMENTAL STUDY

The first step in the implementation of the cryptographic module of secure data transfer was the selection of the necessary operating environment, so the Visual Studio 2019 development environment was chosen to create and develop the application, this product contains an integrated software development environment and a set of tools that will be needed during the experimental study.

To implement all the technical tasks of the experimental study, the high-level programming language C++ was chosen, this language provides all the necessary tools for the successful completion of the task and is the optimal solution for organizing work with OpenSSL cryptographic libraries. The cross-platform software development toolkit Qt Creator version 5.0.2 (Community) was chosen to create the application’s window interface. This product contains several basic elements that are necessary for creating application software, starting with graphical interface elements, and ending with classes for working with the network. The last software product to be used in the study is the universal OpenSSL library. This product is

a universal set of libraries for working with cryptography and is used to work with Secure Sockets Layer and Transport Layer Security protocols.

Since the exchange process takes a fairly short period of time, to demonstrate more stable results of the proposed software module, it is necessary to establish the connection and exchange messages several times in a row and display the result in the same table with the minimum, average, maximum values, as well as the median and 90 and 99 percentiles. Logging of the session encryption key to a file in the NSS Key Log Format is also provided, with the ability to use it to view decrypted traffic in Wireshark. Block diagram of the proposed cryptographic module for protected data transfer using the improved TLS protocol by means of OpenSSL a presented on Fig.1.

Testing the software module for establishing protected data transmission is performed in several stages. First, the main window generates the DSTU 4145-2002 certificate. After successful certificate generation, an inscription about the successful creation of certificates is displayed at the bottom of the program, namely ca-cert, server-cert, ca-key, server-key, server-req. All the files listed above have a PEM extension. These files will be needed to establish a secure connection between the client and the server, respectively ca-cert and server-cert are certificates of the certification authority and the server, ca-key and server-key are their keys, and server-req is a CSR request.

The root certificate is generated first, the root certificate is signed by itself using the generated private key. The private key must be stored in a safe storage, and having it, any certificate can be signed on our behalf. Obtained root certificate can be used to identify the authenticity of the server certificate.

To sign a server certificate, it is necessary to generate a key and a signature request, and then send a CSR request to the certification authority. After generating all the necessary files, the server sends the client its digital certificate (signed by a certification authority) and its public key. In our case, it is necessary to select the session encryption algorithm and click the start server button. After that, the server will wait for the client to start.

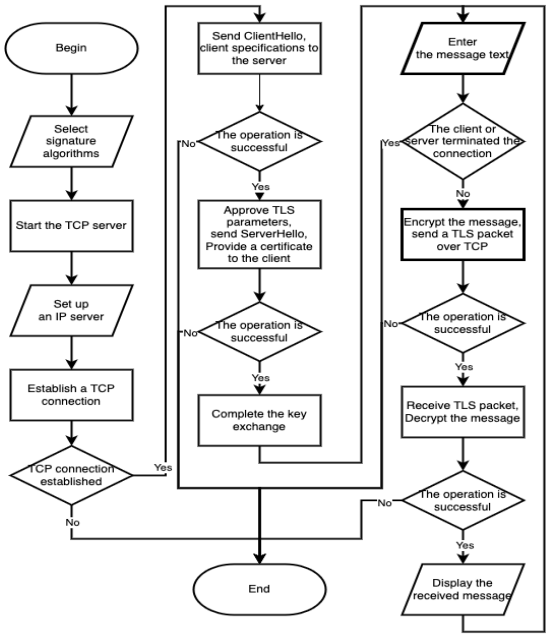

Figure 1. Block diagram of the proposed cryptographic module for protected data transfer using the improved TLS protocol by means of OpenSSL

ISSN 2663 - 4023

After launching the server, the client needs to be connected, to do this, enter the required IP-address and connect to the server. A session key for a secure connection is generated, this is done as follows: a random digital sequence is generated, the client encrypts it with the server's public key and sends the result to the server. The server, in its turn, performs the decryption procedure using the private key. Since the encryption algorithm is asymmetric, only the server can decrypt the sequence since it is the only one with the private key.

As a result of the performed actions, a secure connection was established for data transfer between the server and the client. Immediately after establishing a secure connection, a session encryption key will be generated in the keylog.log file, which will contain encrypted traffic during the connection between the client and the server. If the certificate is unauthorizedly modified, a number of errors will be displayed, and the connection between the client and the server will not be established.

To view the certificate request, let's use the PowerShell extensible shell in which we will enter a command from the OpenSSL library: openssl req-in server-req.pem-noout-text (Fig. 2).

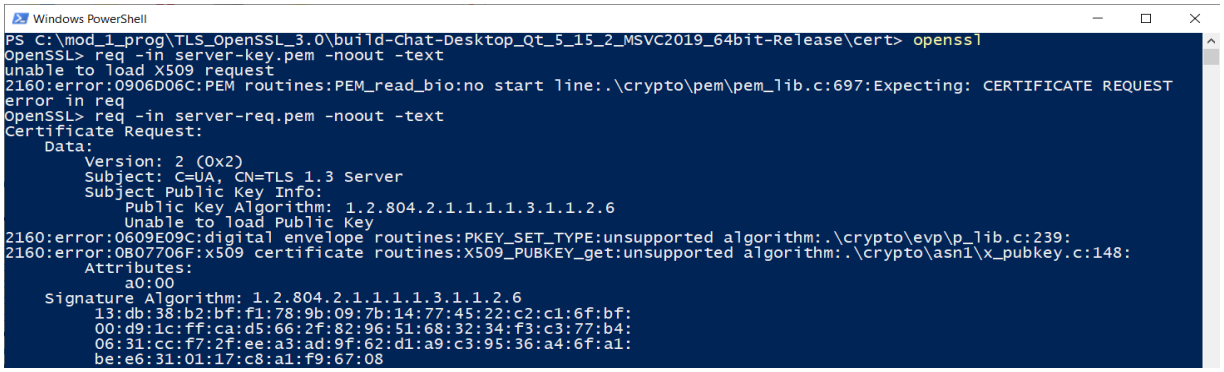

Figure 2. Contents of the server-req.pem file

To view the necessary information about certificates, let's use the PowerShell extensible shell in which we enter a command from the OpenSSL library: openssl x509—in (ca-cert.pem/server-cert.pem)—text. To decrypt TLS traffic, we will use the Wireshark network packet analyzer program. While our cryptographic module is running, let's capture traffic through Wireshark using the Adapter for loopback traffic capture utility (Fig. 3–5).

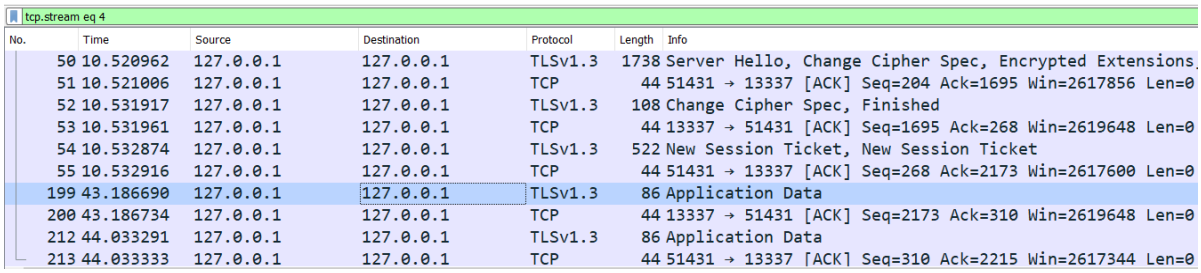

Figure 3. Captured TLSv1.3 traffic

After capturing the traffic, we need to sort the received traffic. Let's select TCP traffic using the dialog filter and find our messages.

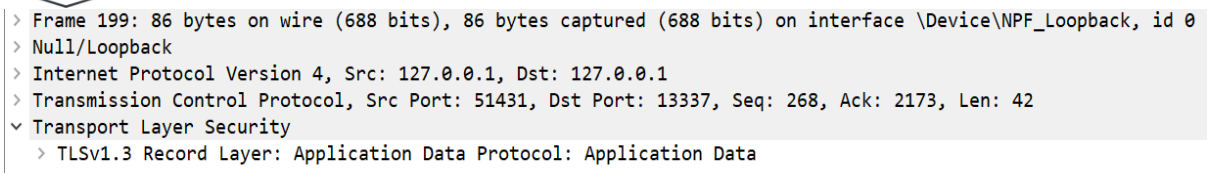

Figure 4. Information about the packet where the message is contained

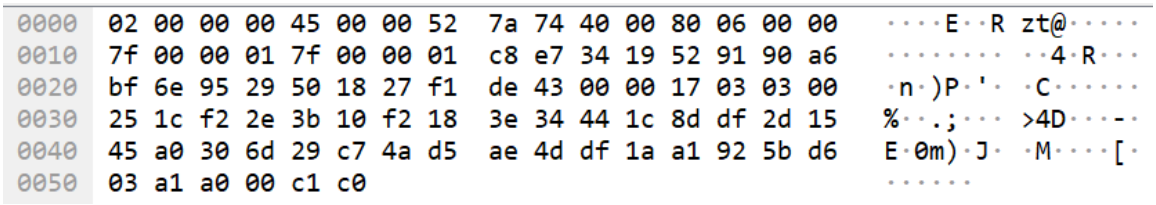

Figure 5. Encrypted packet

To perform decryption, let's use the session key that was created after establishing a secure connection between the server and the client. In Wireshark, go to Edit—Settings—Protocols and select TLS. In the TLS window, specify the keylog.log file (Fig. 6).

As a result of the performed manipulations, we will get a set of messages that was intercepted by Wireshark software.

The decryption was performed to demonstrate the correctness of our cryptographic module, as we can see from the performed experiment, the traffic transferred during a secure connection between the client and the server can be decrypted only if the session key is available on the server and is securely stored.

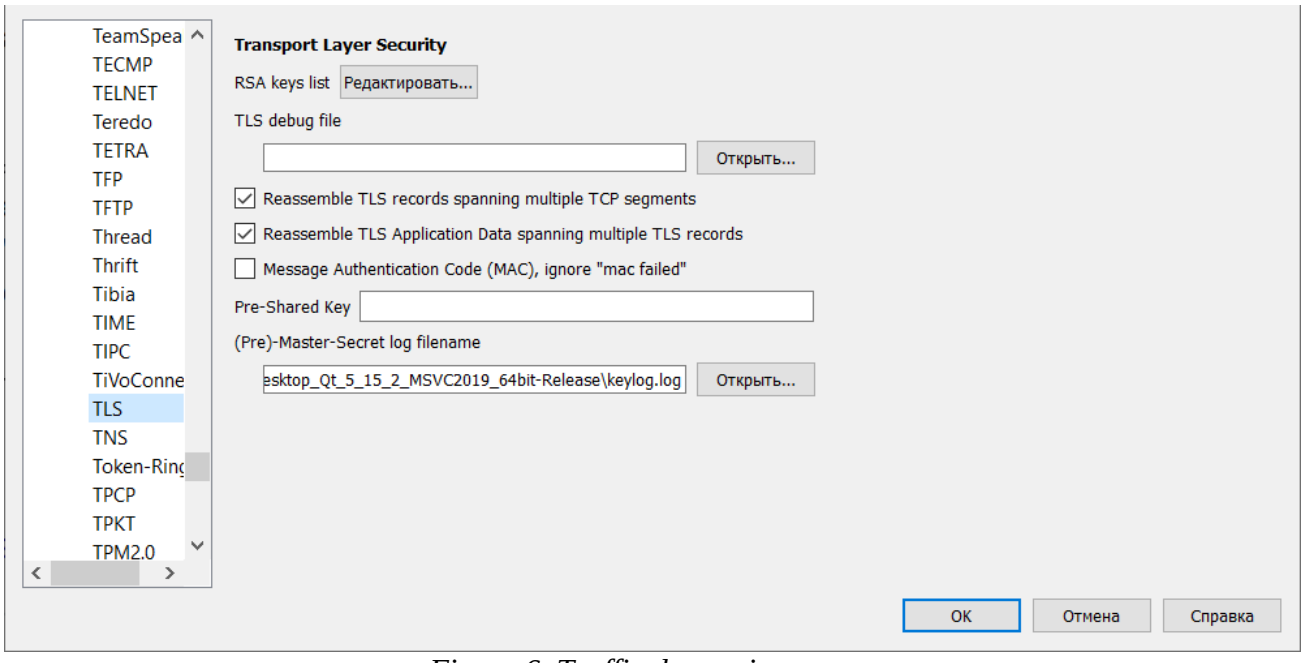

Figure 6. Traffic decryption

Comparative analysis of RSA, ECDSA and DSTU 4145-2002 algorithms.

Let's compare the speed of three digital signature algorithms. To do this, in the software module, we will execute a test that measures the speed of signature creation, verification speed, handshake establishment speed, and full exchange speed per session. For better clarity, let's display the quantitative characteristics of testing, such as: the number of created and verified

signatures, the number of established handshakes, the number of messages per session, and the size of messages per session.

During algorithms testing, let's create a thousand signatures and calculate the minimum, average, median, 90th percentile, 99th percentile, and maximum speed values.

To encrypt the session, let's use the ChaCha20 stream cipher and the Poly1305 Message Authentication Algorithm (MAC), as it is positioned as a faster and more secure analog of AES-256 due to the use of simpler primitives.

During the work, the following experimental studies were carried out:

1) testing the speed of qualified digital signature algorithms using the AES 256 session algorithm with GCM mode and SHA 384 hashing mode. Since the algorithm is accelerated in hardware, the speed value will be higher than in ChaCha20;
2) testing the speed of qualified digital signature algorithms using the AES 128 session algorithm with GCM mode and SHA 256 hashing mode. As can be seen from Table 1, the DSTU 4145-2002 algorithm proposed for use in the TLS 1.3 protocol is much faster in all indicators than the ECDSA and RSA digital signature algorithms. When using the AES_128_GCMSHA256 session algorithm, the highest speed is provided, this is generally due to the extension of the x86 command system for microprocessors, which leads to acceleration of applications. However, if the AES NI function is not provided by the technical support, the CHACHA20_POLY1305_SHA256 algorithm will be faster since it has simpler primitives (Table 1).

CONCLUSIONS

Summarizing the outlined in this article, it can be concluded that the proposed software solution for improving the TLS 1.3 protocol using DSTU 4145-2002 allows to ensure the security of data transfer over open communication channels and protect against unauthorized external influence. The article highlights the theoretical description of the solution for organizing a secure data transfer channel and develops a technical task that describes the components of the cryptographic module for establishing a secure connection between the server and the client for the exchange of information data.

The article proposes a software application for secure data transfer using the improved TLS protocol by means of OpenSSL through the implementation of DSTU 4145-2002 standard for the creation and verification of electronic digital signature, which allows to ensure the authenticity and confidentiality of information using the C++ programming language and allows to improve the performance of the TLS 1.3 protocol up to 4 times by using the DSTU 4145-2002 standard in its architecture and expanding the x86 command system for microprocessors.

Table 1

Comparison of algorithms speed by the 99th percentile

Algorithm	Speed (ms)	CHACHA20_POLY1305 SHA256	AES_256_GCM SHA384	AES_128_GCM SHA256
RSA (3072 bit)	Signature creation	8.418	4.175	10.85
	Signature verification	0.412	0.144	1.084
	Handshake establishing	8.626	5.594	10.769

	Full exchange time per session	0.605	0.524	0.459
ECDSA (NIST - secp256r1)	Signature creation	0.682	0.12	0.179
	Signature verification	1.012	0.196	0.27
	Handshake establishing	7.315	3.574	3.363
	Full exchange time per session	0.565	0.385	0.384
DSTU 4145-2002	Signature creation	0.169	0.141	0.165
	Signature verification	0.361	0.204	0.359
	Handshake establishing	2.438	2.708	2.218
	Full exchange time per session	0.466	0.339	0.313

A practical analysis of the performance of digital signature algorithms for establishing a secure connection demonstrated that DSTU 4145-2002 is several times faster than its predecessors, regardless of the choice of session key. The experiment was based on a thousand iterations of signature creation and verification, where the maximum, minimum, and average values were determined in a table. The minimum speed of the algorithm did not exceed three seconds, the average did not exceed four seconds, and the maximum did not exceed five seconds. All three test results of DSTU 4145-2002 have never exceeded the values of the experiments with RSA and ECDSA algorithms. These methods were tested and the results of protection in the information network during data transfer were shown.

It should also be noted that one of the advantages of DSTU 4145-2002 is that it is approved at the state level, which makes it possible to use the algorithm in state institutions and organizations, and the flexibility of the standard should also be noted, which allows it to be adapted to various software.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Menezes, A., Paul, C., & Scott, A. (2018). Vanstone. Handbook of applied cryptography. *CRC press*.
2. Mollin, R. (2006). An introduction to cryptography. *Chapman and Hall/CRC*.
3. Aumasson, J.-P. (2017) Serious cryptography: a practical introduction to modern encryption. *No Starch Press*.
4. Aumasson, J.-P. (2021). Crypto Dictionary: 500 Tasty Tidbits for the Curious Cryptographer. *No Starch Press*.
5. Mihailescu, M., & Nita, S. (2021). Pro Cryptography and Cryptanalysis with C++ 20: Creating and Programming Advanced Algorithms. *Apress*.
6. Sklavos, N., et al. (2017). Wireless security and cryptography: specifications and implementations. *CRC press*.
7. Sen, J., et al. (2013). Theory and practice of cryptography and network security protocols and technologies. *BoD–Books on Demand*.
8. Devi, T. (2013). Importance of cryptography in network security. *2013 International conference on communication systems and network technologies*.
9. Sarkar, A., Swagata R., & Mohuya C. (2021). Role of cryptography in network security. *The Essence of Network Security: An End-to-End Panorama*, 103–143.
10. Forouzan, B., & Mukhopadhyay D. (2015). *Cryptography and network security*, *12*.
11. Wang, J., & Kissel, Z. (2015). Introduction to network security: theory and practice. *John Wiley & Sons*.
12. Viega, J., Messier M., & Chandra, P. (2002). Network security with OpenSSL: cryptography for secure communications. *O'Reilly Media, Inc*.
13. Acıiçmez, O., & Schindler, W. (2008). A vulnerability in RSA implementations due to instruction cache analysis and its demonstration on OpenSSL. *The Cryptographers' Track at the RSA Conference*, 256–273.
14. Käsper, E. (2012). Fast elliptic curve cryptography in OpenSSL. *Financial Cryptography and Data*

Security, 27–39.

15. Ilyenko, A., & Ilyenko, S. (2022). Program Module of Cryptographic Protection Critically Important Information of Civil Aviation Channels. In *International Conference on Computer Science, Engineering and Education Applications*, 235–247.
16. Kazmirchuk, S., Ilyenko, A., & Ilyenko, S. (2019) Digital signature authentication scheme with message recovery based on the use of elliptic curves. In *International Conference on Computer Science, Engineering and Education Applications*. 279–288.
17. Kazmirchuk, S., Ilyenko, A., & Ilyenko, S. (2020). The Improvement of Digital Signature Algorithm Based on Elliptic Curve Cryptography. In *International Conference on Computer Science, Engineering and Education Applications*, 327–337.

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (22), 2023
ISSN 2663 - 4023

Ільєнко Анна Вадимівна
кандидат технічних наук, доцент, доцент кафедри комп'ютеризованих систем захисту інформації
Національний авіаційний університет університет, Київ, Україна
ORCID 0000-0001-8565-1117
ilyenko.a.v@nau.edu.ua

Ільєнко Сергій Сергійович
кандидат технічних наук, доцент, доцент кафедри автоматизації та енергоменеджменту
Національний авіаційний університет університет, Київ, Україна
ORCID 0000-0002-0437-0995
ilyenko.s.s@nau.edu.ua

Прокопенко Олена Володимирівна
старший викладач кафедри комп'ютеризованих систем захисту інформації
Національний авіаційний університет університет, Київ, Україна
ORCID 0000-0001-9895-888X
bortnik.olena.v@nau.edu.ua

Кравчук Ірина Анатоліївна
асистент кафедри комп'ютеризованих систем захисту інформації
Національний авіаційний університет, Київ, Україна
iryna.kravchuk@npp.nau.edu.ua

ПРАКТИЧНІ ПІДХОДИ ОРГАНІЗАЦІЇ ЗАХИЩЕНОЇ ПЕРЕДАЧІ ДАНИХ ПО ПРОТОКОЛУ TLS ЗАСОБАМИ OPENSSL

Анотація. Захист інформації під час передачі повідомлень є одним із найважливіших завдань в сучасному світі. Робочі станції конкретної організації можуть бути надійно захищені з використанням програмно-апаратних засобів, але при передачі інформації у відкритий незахищений простір виникає велика ймовірність витоку, перехоплення та підміни даних. У більшості випадків використання недостатньо ефективних засобів захисту стає причиною втрати персональних даних громадян, даних підприємства чи організації які мають характер комерційної таємниці, інформації з обмеженим доступом або взагалі державної таємниці сектору критичної інфраструктури. В такому випадку доцільним вважається використання сучасних криптографічних методів для організації захищеної передачі даних. Зважаючи на постійно зростаючу статистику кібератак на інформаційно-телекомунікаційні мережі, після глибокого аналізу та опрацювання зазначеної проблематики, автори статті висвітили сучасний сучасні напрями захисту та рішення щодо безпеки в інформаційно-телекомунікаційних мережах. Автори всебічно охопили та дослідили основні засади сучасного стану захищеності передачі даних та організації захисту інформації під час її передачі за допомогою протоколу TLS, що дозволило визначити напрямки рішення або модернізації вже існуючих засобів захисту інформації. Також приділено увагу розробленню програмної реалізації модуля захищеної передачі даних в інформаційно-телекомунікаційній мережі по удосконаленому протоколу TLS засобами OpenSSL, що дало змогу встановлювати з'єднання за допомогою алгоритмів електронного підпису. Авторами планується ряд науково технічних рішень щодо розробки та впровадження ефективних криптографічний методів забезпечення безпеки інформаційно-телекомунікаційних мереж.

Ключові слова: мережа; шифрування; протокол; електронний підпис; захист інформації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Menezes, A., Paul, C., & Scott, A. (2018). Vanstone. Handbook of applied cryptography. *CRC press*.
2. Mollin, R. (2006). An introduction to cryptography. *Chapman and Hall/CRC*.
3. Aumasson, J.-P. (2017) Serious cryptography: a practical introduction to modern encryption. *No Starch Press*.

4. Aumasson, J.-P. (2021). Crypto Dictionary: 500 Tasty Tidbits for the Curious Cryptographer. *No Starch Press*.
5. Mihailescu, M., & Nita, S. (2021). Pro Cryptography and Cryptanalysis with C++ 20: Creating and Programming Advanced Algorithms. *Apress*.
6. Sklavos, N., et al. (2017). Wireless security and cryptography: specifications and implementations. *CRC press*.
7. Sen, J., et al. (2013). Theory and practice of cryptography and network security protocols and technologies. *BoD–Books on Demand*.
8. Devi, T. (2013). Importance of cryptography in network security. *2013 International conference on communication systems and network technologies*.
9. Sarkar, A., Swagata R., & Mohuya C. (2021). Role of cryptography in network security. *The Essence of Network Security: An End-to-End Panorama*, 103–143.
10. Forouzan, B., & Mukhopadhyay D. (2015). *Cryptography and network security*, *12*.
11. Wang, J., & Kissel, Z. (2015). Introduction to network security: theory and practice. *John Wiley & Sons*.
12. Viega, J., Messier M., & Chandra, P. (2002). Network security with OpenSSL: cryptography for secure communications. *O'Reilly Media, Inc.*
13. Acıiçmez, O., & Schindler, W. (2008). A vulnerability in RSA implementations due to instruction cache analysis and its demonstration on OpenSSL. *The Cryptographers' Track at the RSA Conference*, 256–273.
14. Käsper, E. (2012). Fast elliptic curve cryptography in OpenSSL. *Financial Cryptography and Data Security*, 27–39.
15. Ilyenko, A., & Ilyenko, S. (2022). Program Module of Cryptographic Protection Critically Important Information of Civil Aviation Channels. In *International Conference on Computer Science, Engineering and Education Applications*, 235–247.
16. Kazmirchuk, S., Ilyenko, A., & Ilyenko, S. (2019) Digital signature authentication scheme with message recovery based on the use of elliptic curves. In *International Conference on Computer Science, Engineering and Education Applications*. 279–288.
17. Kazmirchuk, S., Ilyenko, A., & Ilyenko, S. (2020). The Improvement of Digital Signature Algorithm Based on Elliptic Curve Cryptography. In *International Conference on Computer Science, Engineering and Education Applications*, 327–337.