UDC 004.056.55

Y. Kornaga, V. Garmatin, A. Hryshko, A. Maksimyuk, V. Gasanov

PROTECTION OF AN ELECTRONIC DOCUMENT USING A CONSOLIDATED APPROACH TO THE APPLICATION OF ELECTRONIC DIGITAL SIGNATURE

Abstract: The article deals with the use of electronic digital signature based on software models of cryptographic methods for protecting the document integrity. The most appropriate in terms of protection software tools for working with electronic digital signature were analyzed. There was defined advantages and disadvantages of the algorithms. An algorithm for initiating and authenticating a user that uses an electronic digital signature was proposed.

Keywords: network infrastructure, document, electronic document, electronic digital signature, authentication, verification, EdDSA, SHA-512, Argon2.

Introduction

The Internet is an integral part of our everyday life, and therefore a guarantee of a high level of security is one of the priorities in the development of information technology. Currently, users and the Internet infrastructure, such as routers, servers and services, are targets of various kinds of malicious attacks, such as denial of service attacks, hacks, phishing attacks and unwanted e-mail (spam) - especially dangerous - and ensuring sufficient protection indicator involves the effective use of the Internet.

The reason of the vulnerability of the Internet to various attacks lies in its initial creation goals, in which it was assumed that the network would be used in a completely different way than today. Initially, the Internet was developed to be used by relatively few friendly parties. At present, the situation is quite different: a large number of different users are using the Internet, and almost all attacks against it occur from within the network. Protecting the Internet from attacks is quite a difficult task, since there is no effective security measure that can cover all types of attacks. You can protect yourself as much as possible against direct attacks from the network using HTTPS (TLS1.2 + trusted certificate), but you should be afraid of various attacks based on social engineering. Unfortunately, only awareness and attentiveness of users can save him from this type of attack.

Problem statement and current developments

Traffic verification and integrity check can be provided for comprehensive security solutions. One of the main problems is the low efficiency: if the network infrastructure compromised and cannot deliver packets. This requires a clear need for new solutions. If traffic can be verified in the network infrastructure, control measures can be taken on the network. This will allow you quickly and effectively stop the attack and keep your data safe.

Document management systems typically provide security and control access to documents in a controlled environment. However, when a document leaves a secure environment, it is easy to modify it. Unprotected documents do not allow

to determine whether the document is authentic, who was the originator and the approver or has it been modified since its creation.

The problem of keeping electronic documents from being copied, modified, and forgery requires specific approaches and methods of protection for its solution. One of the most common method of such protection in the world is an electronic digital signature (EDS), which confirms the authenticity, integrity its details and the fact of signing by a specific person of the document with the help of special software.

Dependence on digital signatures alone is a matter of concern, since a pair of keys can be obtained by another person or organization using one method or another. This can be resolved by verifying by certification center. The certification center is a trusted third party (for example, a bank) that will ascertain the identity of the person or company. For example, it can be done by checking passports or driver's license details, as well as corporate documents. Then certificate center will issue a digital certificate signed with its own digital signature, which will be attach to user’s digital signature as an identity card.

Certification center is a trusted third party that provides information about the identity of the key holder in the form of an authenticated key certificate [2]. All electronic certificates are digitally signed by a certification authority with a private key. If the certification authority supports strong private key protection, it is almost impossible to forge an electronic certificate. The certificate can be distributed in several ways. The certificate can be “handed over” to the owner of the signature. Then the owner can distribute the certificate anywhere he decides. This approach is preferable to publishing a certificate on a website.

Modern information systems allow organizations to improve their efficiency, significantly reduce their costs and meet regulatory requirements. A good document management system is often regarded as all that is required, but additional protection measures are also needed to ensure that data is protected from unauthorized access and forgery.

The purpose of this article is to modify the algorithms of cryptographic methods of protecting the integrity of a document using an electronic digital signature.

Use of electronic signatures creates significant problems in relation to the individual. The use of paper tools for creating and maintaining records often includes handwritten signatures, and verification tools such as seal are the predominant approach of performing official actions. Typical examples of paper rules are formal legal requirements in favor of paper documents and handwritten signatures or archiving rules that require storing valuable information on paper. These rules can be found in various national, international and supranational legal frameworks.

Traditionally, a handwritten signature is a sufficient means of authentication. By signing a paper document, the manufacturer “identifies” itself as the author of the document and confirms the “integrity” of the document. The signing procedure serves as a warning, and also confirms the fact that the information has been finalized and was

 ISSN 1560-8956

not changes since signing. Distinguishing marks can be encoded in the information itself to identify the source and authenticate the content. Many forms of digital authentication are currently used, such as using a password, such as a PIN code, using encryption methods such as digital signatures, and using biometric identification, such as finger-prints, face, retina and voice recognition. Basically, these authentication methods are combined to provide a high level of authentication security.

The issue of identification is a concern due to insufficient data protection. However, the European Directive requires general compliance of the Data Protection Directive (95/46/EC) [13]. The Electronic Signature Directive also requires Member States [13] to ensure that service providers issuing certificates to the public do not collect personal data, except directly from the data subject, or without the explicit consent of the data subject. There is another requirement that only required data in case of necessity can be collected for the certificate maintenance and issuance.

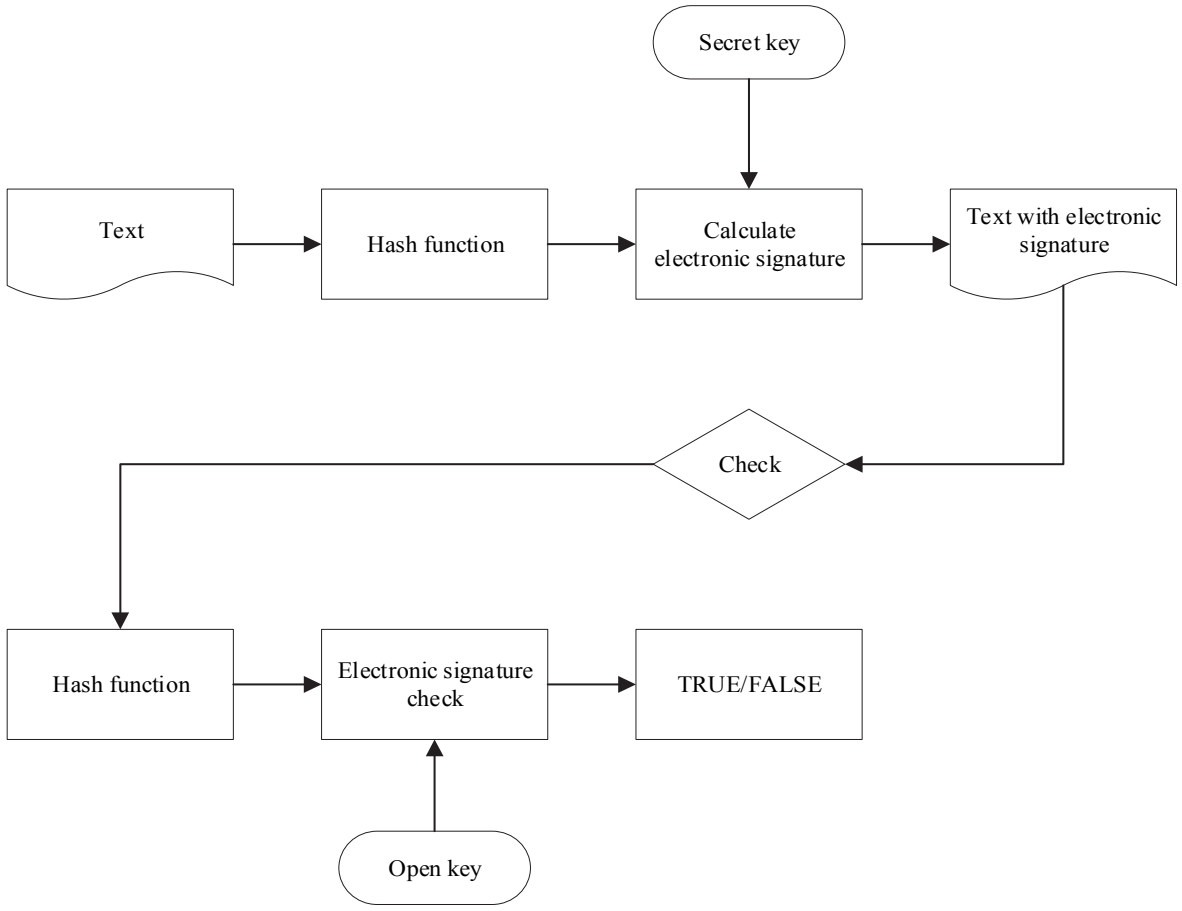

Fig. 1. Typical Electronic signature algorithm

There are several schemes of constructing a digital signature [1]:

- Based on symmetric encryption algorithms. This scheme provides for the presence in the system of a third party enjoying the confidence of both parties. The authorization of the document is the fact of encrypting it with a secret key and transferring it to the authorized person;

- Based on asymmetric encryption algorithms. Currently such schemes are most common and are widely used;

Combined.

Symmetric encryption. Encryption, in which the same cryptographic key is used for encryption and decryption. A huge advantage of this approach is the encryption speed, but both parties, between which information is sent, must know the key. There are two types of this encryption type: stream ciphers, where each character of the message is encrypted one by one with the corresponding digit of the key stream (RC4, SCALA20, Achterbahn); block ciphers that devides the message into blocks of fixed length and operates with the received blocks (DES, AES, Blowfish).

Asymmetric encryption. Encryption, in which there are two types of keys, public and private key. The public key is transmitted over the open channel and is used to check the electronic digital signature and to encrypt the message. The private key is used to generate the EDS and to decrypt the message. Types of asymmetric ciphers: RSA, DSA, ElGamal, Rabin.

Combined encryption. With the help of symmetric encryption, the necessary data is encrypted quickly, the key is attached to the message or the EDS, and the key of the symmetric cipher is encrypted with an asymmetric public key.

For generation EDS, a hash function is needed, which will reduce a message of any size to a certain number of bytes. The resulting hash is encrypted and added to the original document for verification [2]. According to the obtained composition, it is possible to prove that the document has not been changed since the moment of signing.

Common encryption methods are based on factorization of large numbers (RSA, DSA, ElGamal) and discrete logarithmization (ECDSA, EdDSA, GOST R 34.10–2012).

Ed25519 is a type of signature algorithm based on the elliptic curve Curve25519, belonging to the EdDSA family. It was firstly introduced in 2007 as scientific research and from that time it have been spreading among many spheres and companies. These days it is being used by OpenSSH, Apple, Sony, I2p, Tor, Tox, etc. The curve looks like this [11]

$$y^2 = x^3 + 48666 * x^2 + x \quad (1)$$

This is the Montgomery curve over the prime field modulo a prime number $2^{555} - 19$ (which gave the name to the scheme) and with a base point $x = 9$. The scheme uses points in compressed form (only X coordinates), thus allowing the use of the “Montgomery Staircase”, which multiplies the points in a fixed time, saving us from time attacks. Ed25519 consists of three modules [11]:

- Digital signature algorithm;
- Hashing function SHA-512;
- Random number generator for generating key pairs.

ECDSA and EdDSA require the generation of a random value (scalar pair of ephemeral keys) during the signature generation process and the secrecy of this random value is critical for security [12]: knowing one such random value or partial knowledge of several of them allows the signer's private key to be recovered [8].

 ISSN 1560-8956

ECDSA does not describes how to generate this random value and, therefore, implementations critically rely on the quality of the random number. EdDSA removes this dependency by deterministically extracting the secret from the message and the long-term auxiliary key using the SHA-512 cryptographic hash function.

EdDSA is considered to be more resistant to side-channels attack. The authors rely on the idea of “generating random signatures secretly in a deterministic way”, so that “different messages lead to different, difficult-to-predicted ephemeral key values. A few bits can usually be obtained from side-channel attacks or from uneven distribution from which r is taken, so EdDSA authors rightly point out the fact that the “deterministic feature” does not lead to obvious leaks when attacking side-channels attack.

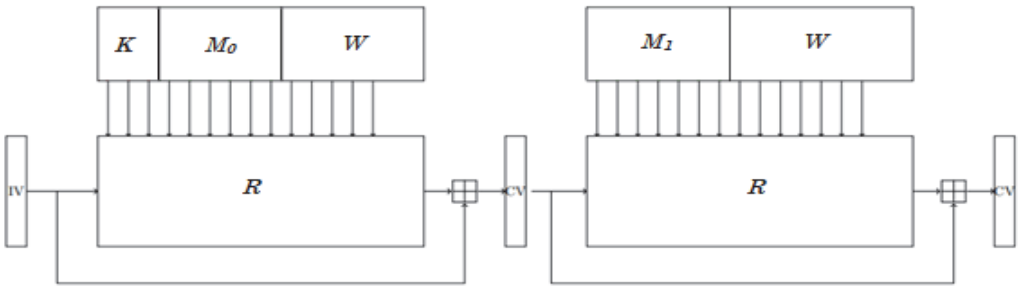

Fig. 2. Generation of the ephemeral key

SHA-512 belongs to the family of SHA-2, developed by the NSA and it has Merkl-Damgard structure [4]. The structure assumes an iterative update of the CV value; this value is initialized with a fixed initial value IV. The message is supplemented (if necessary) and divided into blocks. At each iteration, a message block is processed. The figure shows the generation of an ephemeral scalar, where the auxiliary key and the message are hashed [12]. The letter K denotes the auxiliary key b, M_i - is the input message, W is the queue of the remaining messages, and R is the compression function. M_0 - is a message fragment that is in the same block as the key. Unfortunately, this structure is also vulnerable to side-channel attacks.

It will be enough just a few hundred EDS to find out the auxiliary key, and with it, you can already extract the private key. Although it is possible to protect against such attacks by setting up a random number generator to provide same number sometimes, that you can use same number to sign several messages, this contradicts the security requirements of EdDSA.

If you use electronic digital signature to protect company data, you can modify Ed25519 with another hash function that will vulnerable to the same attacks as SHA-512.

At present, one of the most reliable and flexible functions is Argon2. It was designed for efficient hashing, especially password hashing. It was developed for high speed memory filling (high bandwidth) and efficient use of several computational units (CPU cores), and at the same time providing protection against many types of attacks. Argon2 has three types: Argon2i, Argon2d and Argon2id. Argon2i uses data-independent memory access, which is preferred for hashing, but it is slower as it takes more memory passes for protection against compromise attacks [11].

Argon2d works faster and uses data dependent memory access, which makes it resistant to attacks using the GPU (brute force) and is suitable for applications that do not have to face side-channel attacks [11]. Argon2i reliably fills memory, using 2 CPU cycles per byte, and Argon2d is three times faster [9]. Argon2id is a hybrid of Argon2i and Argon2d, it uses combination of data-dependent and data-independent memory access [11], which gives Argon2i’s some resistance from side-channel attacks and most of Argon2d's resistance from GPU relying attacks.

Despite its high performance, Argon2 provides a reasonable level of reliability and protection. With the default number of passes over memory memory (1 for Argon2d, 3 for Argon2i), an attacker, equipped with an ASIC, cannot reduce the execution time (time-area product) if the memory is 4 or less times less than required [9]. The more memory passes, the more severe penalties will be imposed.

Table 1

Penalties for reading while reducing the required amount of memory

Memory fraction	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{6}$
1 pass	1.7	3	6.3	16.6	55
2 passes	15	410	19300	2^{20}	2^{25}
3 passes	3423	2^{20}	2^{20}	–	–

Table 2

Penalties for time while reducing the required amount of memory

Memory fraction	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{6}$
1 pass	2.7	3.5	4.8	6.7	9.2
2 passes	6.7	13.3	27.8	48	74
3 passes	21.7	57	104	–	–

Argon2 scales both in time and in memory. Both parameters can be changed independently, but you should remember that it always takes a certain amount of time to fill the memory.

Argon2 can use up to 2^{24} threads in parallel, but 8 threads use maximum available bandwidth and computing power of an average PC.

Table 3

Characteristics of the average PC

OS Version	Windows 10 x64
System RAM	8 GB
The number of physical cores	4
Clock frequency	3.3 Ghz

 ISSN 1560-8956

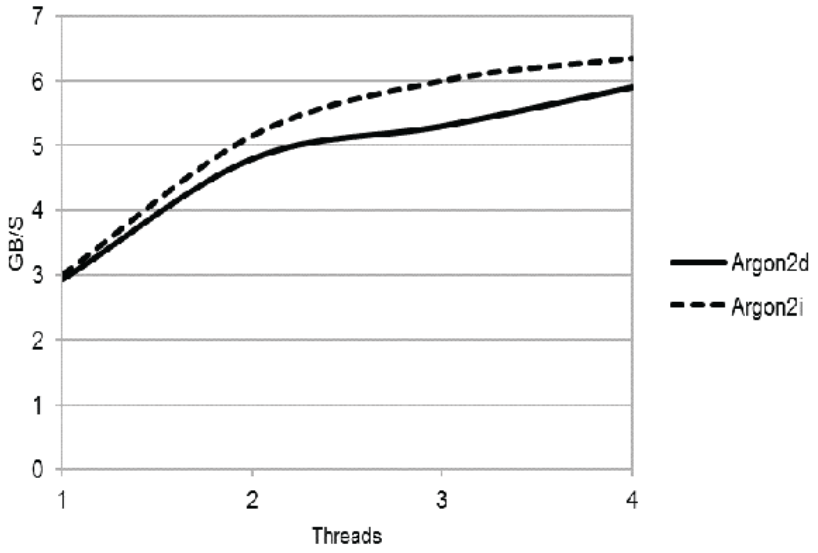

Fig. 3. Memory bandwidth depending on thread

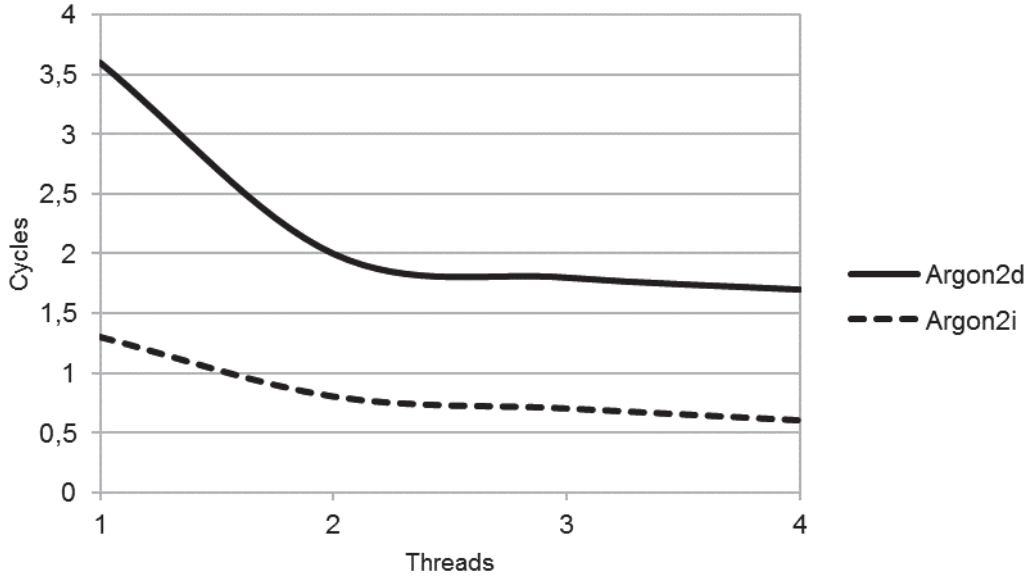

Fig. 4. Performance in cycles per byte depending on thread

Argon2 is optimized for x86 architecture, so its adoption to special equipment will not be either cheap or fast. Even specialized ASICs will require significant space and will not allow decreasing use of time (product of time).

Also Argon2 supports additional input data that is separated from the message and nonce, such as the private key, environment settings, user data, etc.

First, Argon2 hashes the message using the Blake2b hash function. The hash result is written to memory blocks, which are converted using the G compression function (it takes two 8192-bit blocks as input, and outputs a 1024-bit block), and the key is generated as a result.

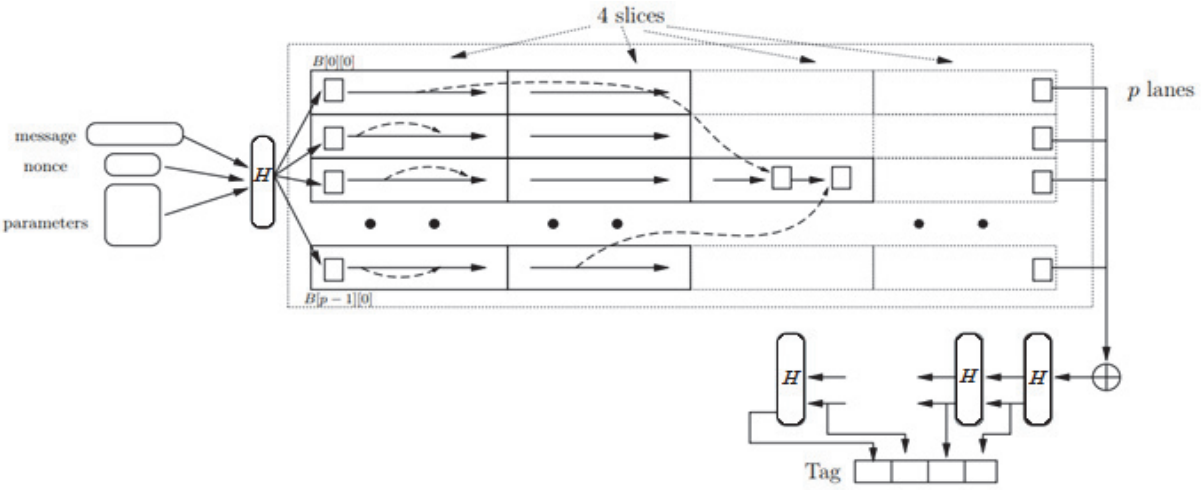

Fig. 5. Argon2 1 memory pass

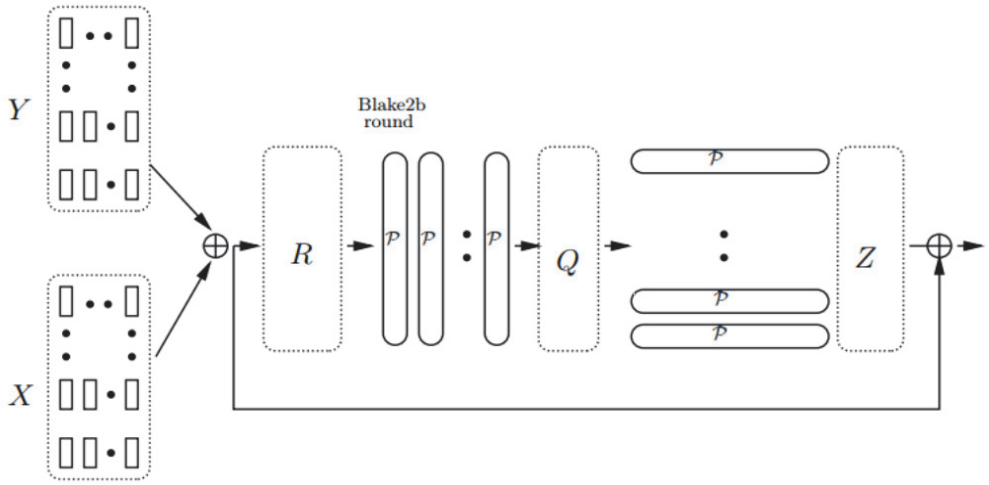

Fig. 6. Compression function

As Ed25519 consists of three of three weakly connected blocks, we can easily swap SHA-512 to Argon2. Thanks to Argon2, you can secure the private key and whole EDS without increasing required computing power, and thanks to Curve25519 the signature size will be only 512 bits.

Conclusion

As a result, it is worth noting that electronic digital signatures have created a completely new level of convenience and secureness for companies all over the world that have dealt with legally binding contracts. What previously was estimated as week for delivery of the contract to the recipient, and then provided the same amount of time to return it signed, now happens instantly. Digital signatures are also much more efficient than sending documents by fax and sending them by courier with a signature, sending documents by e-mail, printing them and scanning signed documents back to a computer.

An electronic digital signature associates a digital sequence with an electronic document as a handwritten signature on a printed paper document. This digital signature should be considered as a handwritten signature. At the core, electronic digital signature is based on using of two different digital keys, known as a pair of keys and two different cryptographic actions. Each pair of keys consists of a private key and a public key. They are interdependent, but can be used separately. Usually, each pair of keys may belong to a specific key holder. The algorithm works in such a way that it is impossible for third parties to calculate the private key, even if they own the public key.

While technology and globalization are growing, electronic digital signatures have become an important requirement for all kind of businesses. In connection with the growing usage of the Internet as an acceptable and truly standard means, there is an urgent need to confirm the growing need for electronic signatures, and therefore research in this area continues, and it is assumed that it should be focused on the need to improve security measures.

REFERENCES

1. EDS — electronic digital sinature. Available at: https://cryptoworld.su/ /ecp-elektronnaya-cifrovaya-podpis-polnyj-manual/ (accessed 23 December 2016);

2. Public-key signatures. Available at: https://libsodium.gitbook.io/doc//public-key_cryptography/ public-key_signatures (accessed September 2018);

3. Algorithm Identifiers. Available at: https://tools.ietf.org/id/draft-ietf-curdle-pkix-06.html (accessed 12 September 2017);

4. Descriptions of SHA-256, SHA-384, and SHA-512. Available at: http://www.iwar.org.uk/comsec/resources/cipher/sha256-384-512.pdf ;

5. Joseph Migga Kizza, Computer Network Security [Department of Computer Science]. Chattanooga, TN, U.S.A., 2005, 534p.;

6. ECM-literacy class, part 1: electronic workflow and archive. Available at: https://habr.com/ru/company/alee/blog/137407/ (accessed 1 February 2012);

7. Aggregate signature implementation based on GOST 34.310-95 and DSTU 4145-2002. Available at: http://pnzzi.kpi.ua/15/15_p82.pdf (accessed 1 November 2007);

8. Breaking Ed25519 in WolfSSL. Available at: https://eprint.iacr.org/ /2017/985.pdf ;

9. Fast and Tradeoff-Resilient Memory-Hard Functions for Cryptocurrencies and Password Hashing. Available at: https://eprint.iacr.org/2015/430.pdf ;

10. The Password Hashing Competition and Argon2. Available at: https://eprint.iacr.org/ 2016/104.pdf ;

11. Argon2: the memory-hard function for password hashing and other applications. Available at: https://github.com/P-H-C/phc-winner-argon2/blob/master/ /argon2-specs.pdf ;

12. High-speed high-security signatures. Available at: https://ed25519.cr.yp.to/ ed25519- 20110705.pdf ;

13. ICT and communication Directive 95/46/EC. Available at: https://ec.europa.eu/eip/ ageing/standards/ict-and-communication/data/directive-9546ec_en(accessed 23 November 1995).