11. PUBLIC KEY CRYPTOSYSTEMS AND THEIR APPLICATION IN DIGITAL SIGNATURE ALGORITHMS

Glushachenko N.S.

Belarusian State University of Informatics and Radioelectronics
Minsk, Republic of Belarus

Perepelitsa L.A. - Lecturer

This paper provides information about what public key cryptosystems are and how they are implemented. It compares symmetric and asymmetric signature methods, describes the use of open key digital signature algorithms.

Public key cryptosystems (asymmetric cryptosystems) are a class of cryptographic algorithms that use two keys: public and private. A public key can be distributed freely among all users, while a private key must be known only by its owner. Public key cryptosystems are used for data encryption and decryption as well as authentication and digital signing. It should be noticed that asymmetric cryptosystems are mostly used to encrypt small amounts of data, which is associated with high computational costs [1].

As mentioned earlier, public key cryptography involves two keys (Figure 1):

– a public key known to all users. It allows to encrypt a message or verify its author;
– private key, known only to the recipient. It is used to decrypt messages or create signatures.
–

Figure 1 – Asymmetric encryption algorithm

Thus, information can be transmitted by any user based on the public key, but only the recipient can decrypt it based on the private key [1]. In this case, unlike symmetric cryptosystems, there is no need to transfer a key between users, which increases the cryptographic strength of the system. A cryptographic function must be one-way. A one-way function is a function for which obtaining an argument, knowing its value, is not possible in a reasonable amount of time with the current level of computing power [2]. This means that it is easy to turn raw data into encrypted data, but impossible to do the opposite in a reasonable time. For example, it is easy to multiply two large prime numbers by each other, but much harder to decompose their product into two large multipliers. Knowing the cipher, key can only be found out by brute force attack. On this basis, one can determine the vulnerability of this type of cryptosystem: if a hacker has enough computing power, they can break the cipher.

Public key cryptosystems address two key issues:

– key distribution – how to have secure communications in general without having to trust a key distribution center with your key;
– digital signatures – how to verify a message that comes intact from the claimed sender.

One of the solutions to the key distribution problem is the Diffie and Hellman algorithm [3]. It is based on the complexity of calculating the discrete logarithm. The essence of this algorithm is that an attacker cannot break the cipher because the secret keys are not transmitted through an open channel, while knowing the cipher it is impossible to get the key because the cryptographic function is one-way.

At first, users select a large prime number q (for example, Mersenne prime numbers can be used [4]), and such a number a, which will be a primitive element of the finite field GF(q) with q – 1 unique elements, which means that these numbers ($a^1 \bmod q$, $a^2 \bmod q$, …, $a^{q-1} \bmod q$) will be unique and form

a random sequence, which excludes the possibility of brute-force attack. It should be noted that q and a can be intercepted, but this will not affect the efficiency of the encryption.

Then both users choose a secret number M in order to compute C. C is an element of GF(q) at M position. In this case, C is distributed through the open channel, which means that there is a possibility of interception by a hacker. However, in order to decrypt C, they will need to calculate M using formula 1, which is impossible because it is a one-way function.

$$M = \log_a C \bmod q \quad (1)$$

Users can then calculate a key based on the C received from the recipient and their own C. With this method of key distribution there is no possibility of interception because it is not distributed over an open channel.

In addition to secret data transmission, this type of cryptosystems can be used to verify information. If a message can be encrypted with a private key and decrypted with a public key, any owner of the public key can decrypt the original message and therefore verify the authorship of the private key owner. At the same time, the one-way function ensures that the private key cannot be obtained in a reasonable amount of time due to the computational complexity of a brute force attack. There are three features that could be stated as the most important properties of such algorithms: anyone can verify the authenticity of the signature, the possibility of forgery is excluded, the author cannot refuse the signature.

The digital signature algorithm is used to authenticate a document or message. It is used in various fields, including banking transactions, e-mails, etc. A digital signature is created by the sender of a message using his private key and verified by the recipient of the message using the sender's public key.

It should be noted, that a digital signature can be realized based on symmetric algorithms as well, but such algorithms are applicable only for single-use signatures. It is connected with the fact that during verification of a signature the user receives half of encryption keys and later can forge the signature. This leads to the necessity to generate unique keys for each signature.

While forming a digital signature, hashing procedure is used. Hashing is a procedure of compressing the text of arbitrary length into messages of fixed length. Such procedure is connected with the fact that without using hash the signature will exceed the length of the text signed with it. So, it is not the text itself being signed, but its hash. In this case it is impossible to get the original message based on the obtained hash. Hash functions are expected to meet the following requirements: the result of hash function must be significantly different for small changes in the original message, hash function must match each unique message with a unique hash, hash repetitions must be excluded.

When creating a digital signature, the sender should do the following:

1. Calculate the hash of the original message using hash function.
2. Encrypt the hash using a secret key. The result is a digital signature.
3. Form a new message consisting of an original message and a digital signature added to it.

The recipient of the signed message must perform the following actions to verify the authenticity of the signature and the integrity of the received message:

1. Calculate the message hash using a hash function.
2. Using the public key, decrypt the digital signature and obtain the original hash.
3. Compare calculated value with hash value extracted from digital signature. If hashes match, the signature is considered authentic.

Falsification of a message during its transmission is possible when a hacker obtains the secret key or by performing a successful attack against a hash function. Hash functions that are used in actual applications have characteristics that make an attack against a digital signature almost impossible [5].

The use of public key cryptosystems in digital signature algorithms provides a high level of security and protection against forgery. A public key can be freely distributed, but it cannot be used to create a signature, as only a private key can be used for this purpose. In addition, it is not possible to change a message after the signature created without detecting changes in the signature itself [6].

Public key cryptosystems are important cryptographic technologies that provide a high level of security and protection in various areas, including digital signature algorithms. Their use is constantly increasing and becoming a necessity in the world of electronic communication and e-commerce.

References:

1. What Is Asymmetric Encryption & How Does It Work? [Electronic resource]. ─ Mode of access: https://sectigostore.com/blog/what-is-asymmetric-encryption-how-does-it-work. ─ Date of access: 20.03.2023.

2. Schneier, B. Applied Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C / B. Schneier, 1996. – 1027 p.

3. What is the Diffie–Hellman key exchange and how does it work? [Electronic resource]. ─ Mode of access: https://www.comparitech.com/blog/information-security/diffie-hellman-key-exchange. ─ Date of access: 20.03.2023.

4. Algorithms for finding prime numbers [Electronic resource]. ─ Mode of access: https://habr.com/ru/post/468833. ─ Date of access: 20.03.2023.

5. Introduction to Digital Signature Algorithm (DSA) [Electronic resource]. — Mode of access: https://www.makeuseof.com/introduction-to-digital-signature-algorithm. — Date of acsess: 20.03.2023.

6. The State of Hashing Algorithms — The Why, The How, and The Future [Electronic resource]. — Mode of access: https://medium.com/@rauljordan/the-state-of-hashing-algorithms-the-why-the-how-and-the-future-b21d5c0440de. — Date of access: 20.03.2023.