Scientific article
UDC 681.3; DOI: 10.61260/2304-0130-2025-2-112-115

IDENTIFICATION AS A MEANS OF INFORMATION ACCESS

✉ **Labinsky Alexandr Yu.**
Saint-Petersburg university of State fire service of EMERCOM of Russia, Saint-Petersburg, Russia.
✉***labinskyi.a@igps.ru***

Abstract. The features and disadvantages of the traditional method of user identification on the network and the features and advantages of identification using a one-time password obtained using a security token are considered. The features of hardware and software tokens are considered.

The identification of a user on the network for the use of a software token is discussed in detail, including installing the application on the user's mobile device, registering the user's device on the network server, and entering the PIN code in the software token application. As a result, the user's device generates a one-time password used to log in to the network.

The advantages of using software tokens are noted.

Algorithms for generating one-time codes for software tokens are considered in detail, including a flowchart and the text of the enlarged modules in C#. The steps of the one-time password generation algorithm are described in detail.

Key words: information safety, secure access, user identification, computer networks, hardware security token, software security token, hash function, encryption algorithm, one-time password

For citation: Labinsky A.Yu. Identification as a means of information access // Supervisory activities and forensic examination in the security system. 2025. № 2. P. 112–115. DOI: 10.61260/2304-0130-2025-2-112-115.

Introduction

The security of information stored in a computer network is ensured by a set of requirements and policies imposed on the network infrastructure in order to analyze its operation and ensure secure access to information.

The traditional way to identify yourself on a computer network is to enter your username and password. Their reference values are stored in a special network database. However, this method of identification on the network has at least two significant drawbacks: the password can be forgotten by the user or intercepted, spied on, or picked up by hackers.

Therefore, various hardware and software tools are currently being used to ensure secure access, among which tokens are widely used. They are a means of identifying a user or an individual session on a computer network.

A hardware token is a compact device that provides electronic user identification, as well as remote access to network information resources. It is small in size, can store cryptographic keys in the form of an electronic signature or biometric data, and usually has USB connectors or a Bluetooth interface for transmitting a specified sequence of key characters to a network access system.

Hardware tokens can contain a fixed access password for each authentication session, a one-time generated password with a certain time interval, a one-time asynchronous password, or a password created using public key cryptography.

The software token has an authorization system that is linked to a specific network client, a network session, or a packet of data hosted on the network. The software token authorization system generates an encrypted sequence of characters to accurately identify the network client and determine its privileges.

Let's formulate the problem statement. It is necessary to consider the features of ensuring secure access to information posted on computer networks using security tokens. The topic of the article is relevant, as the number of network attacks and damage from unauthorized access to information are constantly increasing. Therefore, many works have been devoted to the means of ensuring secure access to information [1–8].

Particularities of software tokens

Software tokens ensure that the password of the network client is limited in time. After said time, the password must be changed. Thus, the password is valid for one login to the network. A new password is required for each subsequent network access request..

The identification of the user on the network is as follows. The application is installed on the user's device as a software (virtual) token. The user then registers his device on the network server. After the user enters the PIN code in the software token application, the user's device generates a one-time password used to log in to the network.

The software token application creates a one-time password using the following information: the current time with an accuracy of one tenth of a second, a four-digit PIN code and a hexadecimal code created when registering the device in the system. Next, the software token application processes this information using the MD5 encryption algorithm, which creates a 128-bit hash value from the input information.

The advantages of using software tokens are as follows:

– the hardware token can be replaced by any mobile device, for example, a mobile phone;

– the program token works on all mobile devices that support java applications (iPhone, Nokia, Siemens, Tesla, Sony, etc.);

– the received one-time password is valid for one network session.

Algorithms for generating one-time codes

The algorithms for generating one-time codes are based on a message identification code using hash functions (HMAC – Hash-based Message Authentication Code). The algorithm for generating a one-time password contains the following steps:

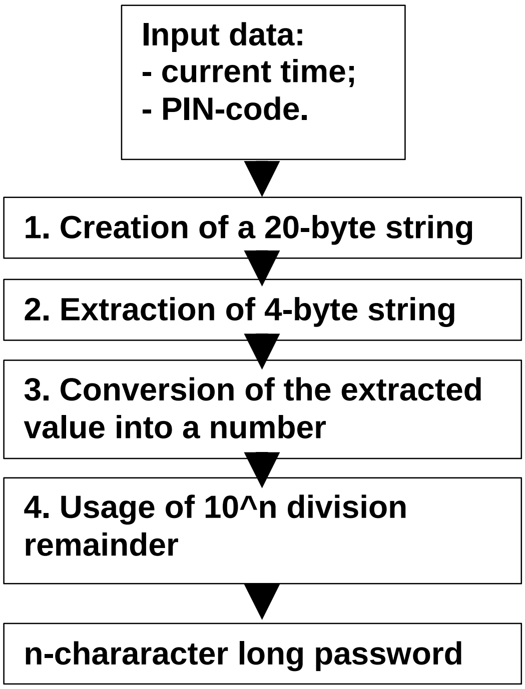
– creation of a 20-byte string based on input information using a hash function;

– extraction of 4-byte information string from a 20-byte string;

– conversion of the extracted value into a number that is divisible by 10 to the power of N, where N is the number of characters in the one–time password;

– usage of the remainder of the division as a one-time password.

The block diagram of the one-time password generation algorithm is shown in the figure.



Input data:
- current time;
- PIN-code.
1. Creation of a 20-byte string
2. Extraction of 4-byte string
3. Conversion of the extracted value into a number
4. Usage of 10^n division remainder
n-chararacter long password

Fig. **The block diagram of the password generation algorithm**

The enlarged algorithm of the C# program in the form of separate modules is presented in tables 1–5.

Table 1

The enlarged text of the module in C#

«const Crypto = window.crypto.subtle;
const encoder = new TextEncoder('utf-8');
const secretBytes = encoder.encode(secret);
const key = await Crypto.importKey('raw',
secretBytes,
{ name: 'HMAC', hash: { name: 'SHA-1' } },
false, ['sign']);»

Table 2

The enlarged text of the module in C#

« function padCounter(counter)
{ const buffer = new ArrayBuffer(8);
const bView = new DataView(buffer);
const byteString = '0'.repeat(64); // 8 bytes
const bCounter = (byteString + counter.toString(2)).slice(-64);
for (let byte = 0; byte < 64; byte += 8)
{ const byteValue = parseInt(bCounter.slice(byte, byte + 8), 2);
bView.setUint8(byte / 8, byteValue);
} return buffer;}»

Table 3

The enlarged text of the module in C#

« function DT(HS) { const offset = HS[19] & 0b1111;
const P = ((HS[offset] & 0x7f) << 24) | (HS[offset + 1] << 16) |
(HS[offset + 2] << 8) | HS[offset + 3]
const pString = P.toString(2);
return pString;}»

Table 4

The enlarged text of the module in C#

« function truncate(uKey)
{ const Sbits = DT(uKey);
const Snum = parseInt(Sbits, 2); return Snum;
}»

Table 5

The enlarged text of the module in C#

«async function generateHOTP(secret, counter)
{ const key = await generateKey(secret,
counter);
const uKey = new Uint8Array(key);
const Snum = truncate(uKey);
const padded = ('000000' + (Snum % (10
** 6))).slice(-6);
return padded; }»

Table 1 shows the module of variables and constants, Table 2 shows the module for creating a 20–byte string, Table 3 shows the module for extracting a 4-byte string, Table 4 shows the module for converting a string to a number, and Table 5 shows the module for obtaining the remainder of the division.

Conclusion

The features of hardware and software security tokens are considered. Algorithms for generating one-time codes for software tokens are considered, including a flowchart and the text of enlarged modules in C#.

Thus, identification using temporary passwords created by a software token is currently the most reliable way to identify a network user. Using this identification method allows you to eliminate the risks that arise when using standard password identification, and as a result, reliably protect data hosted on a computer network.

List of sources

1. GOST R 51275-2006. Information Safety. The object of informatization. Factors influencing information. General regulations // ELECTRONIC FUND of legal and regulatory-technical documentation. URL: http://www.docs.cntd.ru (date of reference: 24.03.2025).

2. Ananchenko I.V., Musaev A.A. Protection of applications. M.: Proceedings of SPIIRAN, 2013.

3. The concept of a security token for software control devices / A.N. Kokoulin [et al.] // Technical Sciences – from theory to practice. 2016. № 1.

4. Markova S.V. Identification of vulnerabilities in information systems // Fundamental studies. 2022. № 9.

5. Shelupanov A. Authentication. Theory and practice of ensuring secure access to information resources. M.: Hotline – Telecom Publishing House, 2009.

6. Johnson M. New adwanced personal data protection. Wiley Information Technologies, 2016.

7. Joseph, Migga, Kizza. Computer Network Security. Springer Science & Business Media, 2005.

8. Jie Wang; Zachary A. Kissel. Introduction to Network Security: Theory and Practice. Wiley, 2015.

9. Owen Poole. Network Security. Routledge, 2007.

10. eToken USB keys and smart cards. URL: http://www.aladdin-rd.ru/catalog/etoken (date of reference: 24.03.2025).

Information about the article: submitted for editing: 16.04.2025; accepted for publishing: 29.05.2025

Information about authors:
Labinsky Alexander Yu., associate professor of the applied mathematics and information technology chair of Saint-Petersburg university of State fire service of EMERCOM of Russia (196105, Saint-Petersburg, Moskovskiy ave., 149), Phd in technical sciences, associate professor e-mail: labynsciy@yandex.ru, https://orcid.org/0000-0001-2735-4189, SPIN: 8338-4230