Electronic Identification of Persons and Objects – Current Status

Karel Burda

Brno University of Technology, Brno, Czech Republic

Abstract
The article expands the theory of identification of persons and objects with new terms and views. The innovated terminology allows for a uniform description of electronic identification systems, such as building access systems, data service access systems, and searching systems. According to the article, there are three methods of identification: presentation, authentication, and determination. The main focus is on verification methods, which are authentication and determination. The article proposes a classification of verification techniques according to entity type and proving factor type. These include verification using biometrics, passwords, techmetrics, and passkeys. Each of these techniques is formally described according to a uniform model, and its development and current status are presented. From the perspective of multi-factor verification techniques, the article also introduces the concepts of direct, witness and aggregated verification.

Keywords:
Identification, verification, authentication, determination.

1. Introduction

Electronic identification of persons and objects is widely used today and continues to evolve. However, due to high specialization, individual types of identification develop more or less independently (e.g. biometric vs. cryptographic identification). As a result, there is no unified technical terminology, no unified description of electronic identification of persons and objects, and no comprehensive overview of the current situation in the field mentioned. This article addresses the issues mentioned above.

In the following, we will refer to persons and objects as entities, where an object is not only an electronic device (e.g., a computer, robot, drone), but also a product without electronics (e.g., an ID card or paper document). This is because such objects can also be identified electronically. Since the terminology for identifying entities is not currently consistent (see below), it is advisable to define the basic terms first.

The first concept is identity, which in its original meaning referred to the match between the data just obtained about an entity (e.g., a person's surname, place of residence, and eye color; or the serial number, model, and manufacturer of an object) and the data about that entity that are certified by the relevant authority (e.g., in a person's passport or on a device delivery note). However, the term identity is now only used in the sense of certified data about an entity (e.g., [1], p. 147). This certified data is stored in an identity database. A necessary element of every identity is an entity identifier. This is usually a string of symbols that is unique for each entity in a given identification system. In the identification system, it serves as a representative of the entity itself, and in the identity database, it serves as a representative of the identity of that entity.

In connection with identification, an entity is usually considered to be an individual person or object. However, an entity can also be understood as a group of persons or objects with the same characteristics. Examples of such entities are persons from the same country or goods of the same type. This concept of entities is used for statistical purposes, such as the number of persons from one country in the territory of another country or the number of items of a given type sold. However, only the identification of individual persons or objects will be addressed below, as the necessary information about the group entity being monitored can be easily derived from these individual identifications.

The term entity **identification** refers to any method of obtaining the identifier of a given entity. Once the identifier is known, further information about the entity (e.g., a person's rights or the price of goods) can be obtained from the identity database. Identification methods include presentation, authentication, and determination. In **presentation**, the identifier is simply provided by or read from the entity itself. The disadvantage of this method is that the presented identifier may be false. However, when the probability of false presentation is low and the potential damage is insignificant, this type of identification is often sufficient. Examples of such scenarios include controlling vehicle access to a parking lot based on their license plate numbers, or website visitor statistics based on their IP addresses. In this context, it is also worth mentioning that there are scenarios where an entity deliberately presents a false identifier in order to remain untraceable. An example is when an RFID card presents itself with a random identifier ([2], p. 5) to first verify that it is not communicating with a fake verifier.

In other scenarios, incorrect identification can cause significant damage (e.g., an unauthorized person can manipulate a bank account). In such cases, it is necessary to use one of the trusted identification methods, which are authentication and determination. **Authentication** is an identification method in which the entity first presents itself,

Manuscript received June 5, 2025
Manuscript revised June 20, 2025
https://doi.org/10.22937/IJCSNS.2025.25.6.1

and then the correctness of the presented identifier is verified. **Determination**, on the other hand, is an identification method in which the correctness of various identifiers is tested according to established rules until the correct identifier is found or all possible identifiers have been tested. Note that the term "determination" is not used in electronic identification systems. However, it is used in biology as a method for identifying species of organisms [3]. Given that both methods are fundamentally the same, it is natural to use an already established term, "determination" in the field of electronic identification as well.

In simple terms, presentation is the statement of an identifier, authentication is the statement and verification of an identifier, and determination is the search and verification of an identifier. Presentation is mainly used for the identification of objects where security requirements are not high. Authentication is used for the trusted identification of both persons and objects. And determination is typically used for the trusted identification of persons for whom presentation is either impossible (typically searching systems) or not suitable. Most often, presentation is unsuitable due to time delays – for example, typing a person's identifier on a keyboard before entry can cause a queue to form at the given entry point.

It has already been mentioned that terms related to identification are not fixed. In particular, the term identification is often used in biometric identification systems to refer to a method that is called determination here (e.g. [4], p. 596 or [5], p. 62). In everyday communication, however, the term identification is used in a much broader sense. For example, according to Merriam-Webster's dictionary, identification is defined as "to ascertain the identity of (someone or something that is unfamiliar or unknown)" [6]. And according to the National Institute of Standards and Technology (NIST) dictionary, identification is defined as "the process of discovering the identity of a person or item from the entire collection of similar persons or items." [7]. From both of these definitions of identification, it is clear that it applies not only to determination, but also to authentication and presentation. Some authors in the field of biometric identification attempt to resolve this contradiction by referring to the method of determination as recognition (e.g., [8]). However, the term recognition only expresses the ability to distinguish one entity from another and does not include the ability to name these distinguished entities. For these reasons, we will adhere to the terminology used in the previous paragraphs.

2. Verification systems

Identification based on presentation is simple in terms of practical implementation. As already mentioned, an entity either states its identifier or it is read from it. This article will therefore focus only on trusted identification methods which are associated with verification systems. A **verification system** is a system designed to trusted find out the identifiers (and thus the identities) of entities. Entities whose identifiers are to be verified by this system must first be registered. During **registration**, the entity is assigned an identifier, a proving factor is agreed upon, and a verification factor is determined. In this way, a list of all registered entities is gradually created, with the X-th entity in this list proving its identifier i_X using the proving factor p_X. The **proving factor** is anything that can be used as proof that the identifier i_X belongs to the entity in question. This can be a significant characteristic of the entity, such as its appearance or activity (e.g., a person's facial appearance or gait). The proving factor can also be a password stored in a person's brain or a secret value stored in an object (e.g., on a chip card). In the case of electronic verification systems, the **verification factor** v_X is always data. Based on this data, the system verifies whether the entity possesses the proving factor p_X.

It is clear from the above that proving factors can generally take various physical forms and may even be intangible. And since an electronic verification system can only work with data, it is necessary to convert the proving factor into data form. The conversion mentioned is performed by **boundary module** F, which converts the proving factor p_X into **evidentiary data** e_X. We will formally write this conversion as $e_X = F(p_X)$. Typical evidentiary data is data that describes the appearance or activity of an entity, or data that depends on a secret value, such as a password or cryptographic key. In the case of authentication, the boundary module also allows the entity to be presented, i.e., its identifier to be stated.

The complete verification system consists of a boundary module F and a test module T (see Figs. 1 and 2), which are interconnected by a secure transmission channel. The **test module** is usually equipped with a verification list, which is a list of pairs (i_X, v_X), where the value i_X is equal to the identifier of the X-th entity and v_X is the verification factor for this entity. The architecture of the verification system in the authentication variant is illustrated in Fig. 1.

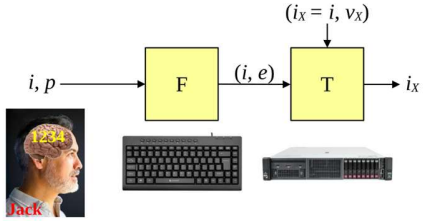

Fig. 1: Architecture of the verification system - the variant with authentication.

In this variant, the boundary module F converts the proving factor p into evidentiary data e and also allows the entity to enter its identifier i into the system in data form. The pair (i, e) is then sent to the test module. Based on the received identifier i, it finds a pair (i_X, v_X) in its verification

list where the value of i_X is equal to the received identifier i, i.e. $i = i_X$. The test module then uses v_X to verify whether the proving factor p_X was used to create the evidentiary data e. If the test result is positive, the verified entity has the proving factor p_X, which belongs to the entity with identifier i_X. This identifier appears as the result in the output of module T.

The lower part of the figure specifies authentication for remote access to data on the server. In this case, knowledge of the text password is often the proving factor. For simplicity, let us assume that this text is also the verification factor. According to the figure, the proving factor p_X is the password 1234 and therefore the verification factor v_X = 1234. The role of the boundary module F is performed by the computer keyboard. With its help, the person enters his identifier i = Jack and creates evidentiary data e = 1234. The pair (i, e) is sent to the server, which in our case also acts as the test module T. Based on the received identifier i = Jack, the module finds the item (i_X = Jack, v_X = 1234) in the verification list. If $e = v_X$, then the proving factor of the person being verified corresponds to the verification factor of the person with identifier i_X. The person being verified is therefore assigned the identifier i_X = Jack.

A special case of authentication is **certificate-based authentication**. A certificate is a data structure containing information that has been confirmed by an authority. In the case of certificate-based authentication, this information consists of a pair (i_X, v_X). During verification, the entity passes its certificate through the boundary module to the test module, which first verifies the authenticity of the certificate. If this verification is successful, the identifier of entity is verified using the pair (i_X, v_X).

The advantage of the described variant is that the verification system does not need to have a verification list for all N entities, but only needs verification data to verify the authenticity of the certificates issued by the relevant authorities. In the case of data certificates, such verification data is the public cryptographic key of the relevant authority.

An example of an identification system where the determination method is used is shown in Figure 2. This figure illustrates an identification system for access to buildings based on a password. In such systems, the person does not present themselves and their proving factor p_X is knowledge of a secret number.

In our example, the proving factor is also the verification factor, i.e. $v_X = p_X$. Let us assume a person with i_X = 12345678, where $v_X = p_X$ = 5858. The verification list therefore contains the record (i_X, v_X) = (12345678, 5858). The role of the boundary module F in the system is performed by a numeric keypad, which the person uses to create the evidentiary data e = 5858. This is forwarded to the access control unit, which also functions as the test module T. The unit then starts searching the verification list for the verification factor v for which $v = e$.

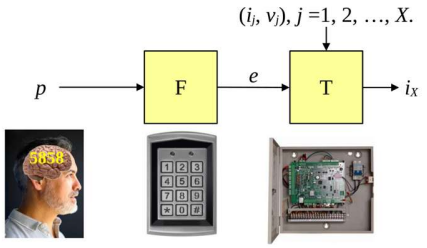

Figure 2: Architecture of the verification system - the variant with determination.

In the example in the figure, a search strategy based on the order of items in the verification list is used. The items (i_j, v_j) are therefore tested sequentially, i.e. j = 1, 2, 3, …, N, where N is the number of items in the list. During the X-th test, it is found that $e = 5858 = v_X$. The search in the list is immediately terminated and the corresponding identifier (in our example i_X = 12345678) is sent to the output of the test module as the result.

In the case of determination, it is necessary to ensure that the verification factors of different entities (in our example, personal passwords) are different. Otherwise, the evidentiary data e could correspond to the verification factors v_j for multiple entities and the identification would then be ambiguous.

Determination is primarily used in searching systems and in building access control systems. In the case of searching systems, the person being searched for will naturally do not present themselves, and therefore authentication cannot be used here. In the case of building access control, presentation is also not applicable because, for example, entering an identifier on a keypad at the entrance to a building would take an unreasonable amount of time. In a typical scenario of access to buildings, where it is necessary to ensure the passage of hundreds of people per hour (typically when employees arrive at work in the morning), such delays would cause unacceptably long queues at the entrances. Therefore, the presentation of persons is not performed and the boundary module only converts the person's proving factor into evidentiary data. The test module T then uses gradual testing to determine who has a verification factor that matches the received evidentiary data, thereby determining the person's identifier. Contemporary verification systems are very fast. For example, a fingerprint verifier [9] can scan four fingers of one hand and check them against a verification list of up to N = 100,000 people in less than 1 second. The throughput of one such pass is then 60 people per minute.

If presentation of entities is possible and does not cause unacceptably long waiting times, authentication is usually used for identification. Its advantage is lower computational complexity, because evidentiary data is tested only once in this case. In contrast, the determination method usually requires more than one test. When the search strategy

shown in Fig. 2 is used, an average of $N/2$ tests must be performed for a verification list with N items.

3. Proving factors

The **proving factor** of an entity can be either secret data stored in the entity or unique and non-imitable characteristics of the entity. These characteristics are most often given by the appearance or activity of the entity. The proving factor p must be converted into evidentiary data e in the boundary module. The conversion of proving factors in the form of secret data into evidentiary data is straightforward, as it is only a conversion of data form or it is calculations. However, the conversion of an entity's characteristics into data is more complex. The general procedure consists of measuring and analyzing the appearance or activity of an entity and then processing the obtained values into a set of numbers. These sets are called **metrics**.

An example where the appearance of an entity is a proving factor is the security features on an ID card. These can be used to verify the authenticity of the card. An example where the activity of an entity is a proving factor is the way a person walks. A person can be identified by the unique characteristics of their gait. An example where secret data stored in an entity is a proving factor is a password. Based on knowledge of the password, it is possible to verify that the person is the correct person.

The verification factor is always data that is usually stored in the verifier. If the proving factor is secret data, the verification factor is either the same data or data that is dependent on the proving factor in a defined way. If the proving factor is the appearance or activity of an entity, the verification data is the relevant metrics.

In cases where the proving factor is the appearance or activity of an entity, it is naturally required that this appearance or activity be not only unique but also inimitable. If the proving factor were imitable, it would be possible to deceive the verifier with other entities or their imitations. If data is the proving factor, it must be kept confidential.

Depending on the type of entity (person vs. object) and the type of proving factor (characteristics vs. data), four basic types of electronic verification can be defined (see Fig. 3).

Types of verification		Type of entity	
		person	object
Type of proving factor	characteristics	biometrics	techmetrics
	data	password	passkey

Fig. 3: Basic types of electronic verification.

If the entity is a person and the proving factor is their characteristics, this is **biometric verification**. Verification of a person whose proving factor is secret data stored in their memory is called **password verification**. If the entity is an object and its characteristics are used as the proving factor, we use the term **techmetric verification** for this verification. For the verification of an object where the proving factor is secret data stored in the object, we use the term **passkey verification**.

Biometric data or passwords are primarily used to verify persons, while techmetric data or passkeys are primarily used to verify objects. We will refer to the above type of verification as **direct verification**. However, trusted entity B can also be used as proving factor to verify the identity of entity A. In this case, entity B acts as a trusted witness who testifies to the identity of entity A. For example, expert B can use their testimony to prove the authenticity of artwork A. Or ID card B can prove the identity of its holder A. We will refer to this type of verification as **witness verification**. In such a case, the verification system generally proceeds by first verifying the identity of witness B and establishing their trustworthiness. If the witness is trustworthy, it then accepts their testimony regarding the identity of entity A. The testimony may be either explicit, i.e. entity B directly states the identity of entity A (for example, the identity of person A is stated on his identity card B), or implicit, i.e. successful verification of entity B (e.g. chip cards) at border module F automatically means that entity A (cardholder) also is present with that module.

Witness verification is most commonly used in electronic identification systems for identifying persons by means of an object such as a passport or chip card. This object is referred to as **proving object**. However, verification by witness carries the risk that an attacker could steal the proving object from the registered person and thereby obtain their identity. Multi-factor verification is used to eliminate this threat. It verifies the authenticity of the proving object and thus the credibility of its testimony. However, this testimony must still be confirmed by some method of direct verification of the holder of the proving object, either by biometrics and/or a password. Using multiple proving factors significantly increases the security and reliability of verification. **Multi-factor verification** generally involves a combination of three proving factors: biometrics, a password, and a proving object. When two proving factors are used, this is known as two-factor verification, and when all three factors are used, it is known as three-factor verification. A well-known example of two-factor verification is withdrawing money from an ATM, where the person must prove their knowledge of the password and also the ownership of the proving object, i.e., the payment card.

Proving object where the proving factor is a passkey must be equipped with suitable hardware for storing and, if necessary, processing this secret value. We therefore refer to this type of object as **proving hardware**. The hardware of these objects can also be used to store digitally signed certificates containing identification and verification data

about a given person. This enables the authentication of persons using a digital certificate. An example is a biometric passport, in which the passport holder's biometric data is stored in a chip. In this case, the verification system, for example at an international airport, reads the certificate, verifies its digital signature, and then uses the biometric data from the certificate to verify the identity of the person. The advantage of this solution is that the border service of each country can verify the identity of any person with a passport from any country. There may be billions of such people, but the verification system only needs a verification list containing the public verification keys of all countries on the planet (approximately 200 entries). These keys can be used to verify the certificate of any country, thereby securely obtaining biometric data to verify the identity of any passport holder on the planet.

Modern proving hardware often disposes an internal verification system – a well-known example is a smartphone with a fingerprint reader. If any external verification system (e.g., a payment system verification system) trusts the internal verification system, then verification using the proving hardware may implicitly include verification of the hardware owner too. An example of this solution is electronic payment using a smartphone. The person must first prove their identity to the smartphone, either by means of a password or biometric data. Only then does the smartphone prove its identity to the payment system. This ensures that witness verification by proving hardware also implicitly includes direct verification of the person. The above witness verification therefore actually represents two verifications. If the internal verification system in the proving hardware performed two-factor verification (i.e., both password and biometrics), then witness verification by the proving hardware against the external verification system would actually represent three verifications. For this reason, we will refer to the described type of verification as **aggregated verification**. The benefit of aggregated verification is that it corresponds to multi-factor verification in terms of security, but the external verification system only has to perform one verification.

In connection with proving factors, it is worth noting their security. Specifically, in the case of metrics, it has already been mentioned that the appearance or activity of an entity should be inimitable. However, inimitability is only relative in terms of technological and knowledge development. What is currently inimitable may be imitable in the future. Fingerprint biometrics is an example of this. Initially, fingerprint sensors were used, where a photograph of the papillary lines of a finger placed on the sensor was taken to prove identity. However, it turned out that the sensor could be fooled, for example, by a gelatin thimble, whose surface lines form the pattern of the fingerprints of one of the registered persons. For this reason, contemporary optical readers are equipped with additional sensors that test, for example, whether blood is pulsing in the object placed on the sensor.

The security issue also applies to secret data. The biggest weakness of this type of factor is that, with the exception of cryptographic verifications, secret data must be transferred to the verifier. If an attacker intercepts this transfer, they obtain the proving factor and thus the ability to deceive the verification system. For this reason, it is necessary to encrypt the transfer. In the case of passwords, another weakness is that as computing power increases, so does the number of passwords that an attacker can test. Therefore, password lengths and the size of the alphabets used must be increased. However, this raises the problem of password memorability. The following chapters describe the different types of direct verification and their security in more detail.

4. Verification by biometrics

In biometrics verification, unique and inimitable characteristics of a person serve as the proving factor. These characteristics are defined either on the basis of appearance or on the basis of the activity of selected organs of the person. Examples of characteristics defined by appearance are specific patterns of fingerprint papillary lines or pigment formations on the iris of the eye. Activity-based characteristics include a person's voice parameters or gait parameters. The appearance or activity of organs is recorded, measured, analyzed, and then expressed as a set of numbers. This quantitative expression of the appearance or activity of organs is called **biometrics**. Biometrics based on appearance are referred to as physiological biometrics, and biometrics based on activity are called behavioral biometrics (e.g., [5], p. 55).

When registering a person with identifier i_x, the system administrator finds out their biometric data. This data is used as the verification factor v_x for that person and is called a template. The pair (i_x, v_x) is then stored in the verification list. In biometric verification, the determination method is generally used to identify a person. In this case, the person has their biometric data determined using the boundary module F, and this data is transferred as evidentiary data e to the test block T. The test block then begins searching the verification list. If it finds verification factor v_x that is sufficiently similar to the evidentiary data e (i.e., $v_x \approx e$), the person is identified as the person with identifier i_x. If an authentication method is used for identification, the person also enters their identifier i_x when the biometric data is captured. The pair (i_x, e) is then sent to the test block. Based on the identifier i_x, the verification factor v_x is used for testing, and if $v_x \approx e$, the person is identified as the person with identifier i_x.

In biometric verification, the criterion for a successful test is not a match, but only a similarity between v_x and e. This is because measuring living organisms is associated

with large errors. Environmental influences are also often involved. For example, in facial recognition, the resulting biometric data depends on the orientation of the person's face, their current facial expressions, the angle of light, etc. As a result, there is a certain probability of errors occurring in biometric verification (e.g., [4], p. 40). The first type of error is when a registered person is declared unregistered. The probability of this error is called the **false rejection rate** (FRR). The opposite type of error is when an unregistered person is identified as one of the registered persons. This probability is called the **false acceptance rate** (FAR). These probabilities are mutually contradictory, because a reduction in FRR leads to an increase in FAR and vice versa. Errors in biometric identification are practically impossible to eliminate due to the dispersion of measurement values, which is a disadvantage of biometric verification.

Biometric identification is currently undergoing rapid development. A number of biometric techniques are commonly used in practice, and many others are being researched and tested. Currently, appearance-based methods are the most widely used. Specifically, these include facial features, fingerprints, and iris patterns (e.g., [10], [11], and [12]). Verification based on the appearance of the vascular network of various organs, such as the palm [13], the back of the hand [14], or the finger [15], is also used. The advantage of vascular patterns is that an attacker cannot easily determine the appearance of this type of proving factor, which complicates any potential attack. In practice, verification based on the appearance of the hand [16] or ear [17] is also used.

Probably the best-known biometric verification method is based on fingerprints, i.e. the appearance of the fingertips. The fingertips are covered with papillary lines, which are ridge lines separated from each other by furrows. These lines create a unique pattern for each person. Figure 4 on the left shows an example of such a pattern with specific features marked that are used for verification.

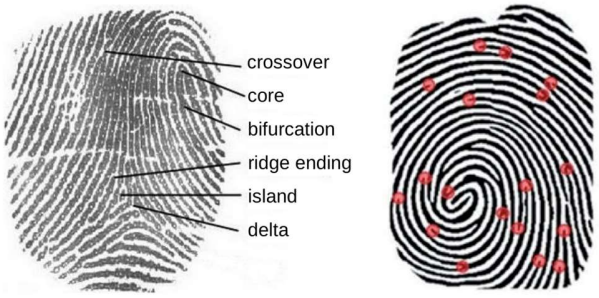

Fig. 4: On the left is a fingerprint where the most common minutiae are marked [18], and on the right is a fingerprint that will be represented in the template by minutiae highlighted in red [19].

These features are called minutiae. The most common are line crossings, the core (the turning point in the narrowest loop), line bifurcations, line ends, islands (very short lines), and deltas (furrow bifurcations). The location and types of minutiae (see red dots in the figure on the right) are unique to each person. A fingerprint template (i.e., the biometric data of a given person) consists of a list of items that contain the type and Cartesian coordinates of each minutia. The person first allows the sensor of boundary module F to scan their finger. The module analyzes the obtained pattern of papillary lines, finds the minutiae, and determines their type and coordinates. This evidentiary data e is then sent to the test module T. The test module tests whether $e \approx v$.

A fingerprint template represents one possible approach to defining biometrics. It is based on a list of features and their parameters. In the case of a fingerprint, these features are individual minutae and their parameters are type and coordinates. Another approach consists in expressing features and their parameters in the form of a single number. We will present this approach using iris verification.

In iris verification, the iris appearance is the verification factor. Figure 5 illustrates the derivation of the verification factor, resp. the evidentiary data of a person.

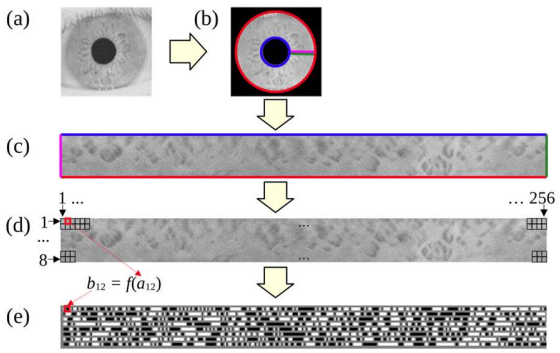

Fig. 5: Procedure for finding the verification factor, resp. evidentiary data during iris verification.

The person's eye is photographed under infrared light, so the resulting image is black and white (a). The iris is selected from the image (b) and its annulus is converted into a rectangle (c) using a mathematical transformation. The resulting rectangle is divided into 8×256 = 2048 elementary areas a_{ij} arranged in eight rows and 256 columns (d). The designation a_{ij} means that the given area is located in the i-th row and j-th column of the rectangle. The pixel brightness in each elementary area (e.g., area a_{12} in the red-framed area) are numbers whose values are used to obtain a single-bit value b_{ij} through appropriate two-dimensional filtering or transformation f. This bit represents the appearance of the entire elementary area. In our example, this is bit b_{12} in Figure (e), where the black color of the elementary area represents a bit with a value of 1 and white represents a bit with a value of 0. The sequence of all 2048 bits from individual areas represents the appearance of the entire iris,

serving as a verification factor v during registration and as evidentiary data e during verification. During verification, the test block performs an XOR operation with the verification factor and the evidentiary data bit by bit, i.e., it calculates the sequence $t = v \oplus e$. The number of ones in the sequence t represents the number of bits whose values are different in the sequences v and e. If this number does not exceed the specified value, the person is successfully identified.

With activity-based methods, identification takes longer because the person must first perform the relevant activity – for example, they must say certain phrases or take a few steps. This is why action methods are not widely used in access systems. On the other hand, however, these methods are very promising for identifying persons in searching systems. Perhaps the most popular method currently is action biometrics based on gait. It exploits the fact that every person has a specific way of walking. A camera is used to record a person's gait, and then the length and width of their steps, their speed of movement, and the various angles formed by the joints in the hip, knee, and ankle in relation to the torso, thighs, calves, and feet are measured and evaluated (see Fig. 6). The described method of identification (e.g., [20]) is very promising for searching camera systems. Another type of action biometrics is based on a person's voice (e.g., [21]). This method exploits the fact that spoken words or phrases have a specific time course of acoustic pressure and a specific frequency spectrum for each person. This type of action biometrics is therefore suitable for use in telephone searching systems. There are also types of action biometrics based on signatures or typing patterns. However, these have not become widely used.

Fig. 6: Examples of models for quantitative description of gaits [22].

A general attack on biometric verification involves an attacker presenting a counterfeit that mimics the registered person's proving factor. In the case of a fingerprint, the attacker can present a thimble made of a suitable material, the surface of which is wrinkled to resemble the registered person's papillary lines (e.g., Fig. 7a). In the case of the iris, the attacker can obtain an acrylic eye prosthesis in the form of a removable shell (e.g., Fig. 7b) or a contact lens. The prosthesis or lens will have an image of the registered person's iris. For face verification, the attacker can wear a face mask in the form of the registered person (e.g., Fig. 7c) or place a photograph of that person in front of the verification camera. In the case of vascular verification, the attacker can place an image of the vascular network of the relevant organ in front of the sensor. For example, Fig. 7d shows a wax cast of a finger with a piece of paper placed in its longitudinal median plane (right). An image of the finger's vascular network is printed on this paper (left). And for voice verification, an attacker can replace the registered person's vocal cords with a suitable reproduction device (e.g., Fig. 7e).

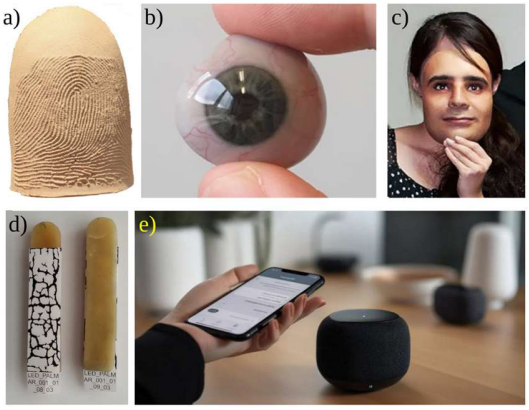

Fig. 7: Attacks on biometric verification. a) Fingerprint counterfeit [23], b) Iris counterfeit [24], c) Face counterfeit [25], d) Finger vein counterfeit [26], e) Voice counterfeit [27].

To protect against the above attacks, when capturing biometric data, it is verified that the carrier of the proving factor is indeed a living person and not an object or device. For example, in the case of a fingerprint, it is verified whether blood is circulating in the finger; in the case of the iris, it can be verified whether the pupil reacts to changes in light intensity; in the case of the face, facial mimics are examined, etc.

5. Verification by techmetrics

In techmetric verification, unique and inimitable characteristics of an object serve as proving factor p. These characteristics are defined based on the appearance or activity of the object in question. Examples of characteristics defined by appearance include patterns printed on an ID card or the surface structure of a plastic card. Activity-based features include transient events in radio transmissions or the state of a series of flip-flop circuits after power-up. The appearance or activity of an object is recorded, measured, analyzed, and then expressed as a set of numbers. For this quantitative expression of the appearance or activity of an object, we will use the term techmetrics.

The principle of operation of a verification system based on techmetrics is the same as that of a system based

on biometrics. When registering an object with identity i_X, the techmetrics v_X of the object is obtained and the pair (i_X, v_X) is stored in the verification list of the system. When an object is verified, the boundary module F to find out the techmetrics of this object and sent to the test module in the form of evidentiary data e. If $e \approx v_X$, the object identifier i_X is confirmed. Unlike biometrics, the results of techmetric measurements have much less variation.

Currently, techmetric verification is most commonly used to verify the identity of objects such as ID cards and banknotes. These items have the character of a paper document on which identification and other data are stated and which is protected against counterfeiting by various security features (e.g., holographic stickers, optically variable inks, etc.). The appearance of the documents is described, respectively verified using their techmetrics.

Signatures were the predecessors of security features for protecting paper documents against counterfeiting. Based on the appearance of the signature, it was possible to verify the authenticity of the document and thus the validity of the information contained therein. Paper documents therefore began to be used as proving factor for identifying individuals. The relevant document (now called an ID card) stated the identity of the holder and was signed by the competent authority. If necessary, the cardholder presented it to a designated person who acted as a verifier (e.g., a doorman or police officer). The verifier compared the signature on the ID card with the signature specimen of the authority and, if they matched sufficiently, considered the cardholder to be the person whose identity was stated on the card. A person's signature is relatively easy to forge, so various security features began to be used to protect documents from forgery. Initially, these were seals, later stamps, and finally special printed techniques. In addition to the similarity of signatures, the verifier also checked the security features on the ID card against those on the card template.

If only the person's identifier is stated on the ID card, it can be stolen and the thief can then assume the identity of the registered person. For this reason, a verification factor for the cardholder began to be included on the ID cards. Initially, this took the form of a text description of the person's appearance, but later a photograph became the verification factor. The person performing the verification task first uses the security features to verify the authenticity of the ID card and then uses the photograph to verify the appearance of the cardholder. The solution, where the authenticity of an ID card is verified using printed security features and the identity of its holder is verified using a photograph, is currently very widespread. The ID card is thus a proving object with certified identification and verification information that allows a human verifier to perform two-factor verification, with the ID card itself and the person's facial appearance serving as the proving factors. The advantage of the described solution is that the authenticity of the ID card and the identity of the person can be verified, for example, by a police officer or civil servant without the need for any technical equipment.

The first electronic identification systems began to appear in the second half of the 20th century. At that time, paper documents with printed security features dominated the field of entity verification. Unfortunately, the electronics of the time were not capable of working with such documents. Therefore, research was initiated with the aim of using electromagnetic phenomena for entity verification. Historically, the first solution was a system for identifying strategic weapons during the Cold War. Each weapon (e.g., a nuclear warhead) was equipped with a label containing a large number of randomly distributed mica grains. The random distribution of the grains was achieved by throwing mica grains onto a sticky layer on the label. This, together with the irregular shape of the mica grains, made such a pattern virtually impossible to replicate. Irradiating the label at a specific angle caused light to reflect, creating a unique pattern that resembled a starry sky. This pattern was used as evidentiary data to identify the weapon in question [28].

Later, other types of reflective protections were designed, this time to verify the authenticity of paper documents. This type of technologies exploited the fact that the surface of every piece of paper or plastic has a unique three-dimensional structure at the microscopic level [29]. Fig. 8 on the left shows a microscopic structure of paper in an area measuring tenths of a millimeter, and on the right is a similar example of a microscopic part of a plastic card measuring tens of micrometers. Irradiating these materials with a laser beam causes reflections of varying intensity at different points on the material. These values are measured and appropriately aggregated into a single number that is unique for a given sheet of paper or plastic card. This number is used as a verification factor during registration and as evidentiary data during verification.

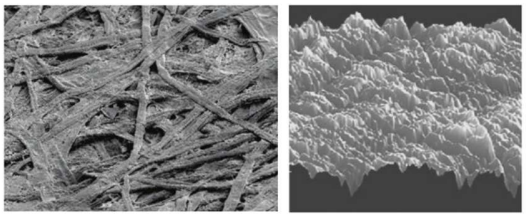

Fig. 8: Microscopic structure of material surfaces. On the left is the surface structure of paper [29] and on the right is the surface structure of a plastic card [30].

In addition to reflective techmetrics, magnetic techmetrics are also known. For example, MagnePrint technology [31] exploits the fact that the magnetic strip on cards is made up of billions of iron oxide nanoparticles. These particles have different shapes and sizes and are randomly mixed during the preparation of the magnetic

suspension. After the suspension dries, each nanoparticle has a random and unchangeable location. Scanning the magnetic field of the strip then creates a unique and inimitable signal, which is used as a verification factor during registration and as evidentiary data during verification.

The development of reflective and magnetic techmetrics looked promising, but at the end of the 20th century, semiconductor image sensors appeared. Together with sufficiently powerful processors, these made it possible to scan and process printed protections, and so reflective and magnetic techmetrics did not become used. Printed techmetrics currently dominate electronic identification systems. They are particularly common in the verification of banknotes and identity cards. The security of this type of verification lies in the fact that exclusive printing technologies are required for a given historical period to create the security features. Some examples of printing techniques that provide protection for contemporary banknotes and ID cards are shown in Fig. 9.

Fig. 9: Examples of printing techniques used to protect documents against counterfeiting. Image a) hologram [32], b) guilloche [33], c) microtext [34], d) optically variable ink [35].

At the top left is a hologram, which is a multi-layered image that creates the illusion of a three-dimensional structure and special color effects when viewed from certain angles. At the top right is an example of a guilloche, which is usually a series of repeating patterns. At the bottom left is an example of microtext, which is text created using very small letters. At the bottom right is an example of optically variable ink, where the color of the image depends on the angle of light. As already mentioned, the above security features can currently only be implemented using exclusive printing technologies [32] and therefore cannot be replicated using commonly available technologies (typically computer printers).

Printed techmetrics of ID cards would only allow single-factor verification by electronic identification systems. To increase security, the modern cards are equipped with hardware that contains identification and verification certificates about the cardholder and also contains a passkey for verifying the hardware. A modern electronic card is now a proving object with certified identification and verification data that allows an electronic verifier to perform three-factor verification, where the proving factors are the card's techmetrics, the person's biometrics, and the passkey of the hardware in the card. An example of this approach is the biometric passport, which has a chip with a microcontroller embedded in it. The chip's memory contains various data that is compared with the printed data (e.g., first and last name, photograph of the person). It also contains the passport holder's biometric data (typically fingerprints or facial metrics). These are then used as verification factors in biometric verification of the person. All of the above data is digitally signed by the relevant authority, and the chip also disposes a so-called private chip key, which can be used to cryptographically verify the authenticity of the chip.

Figure 10 shows an example of a passport page on the left and the output of verifying this passport using a computer on the right.

Fig. 10: On the left, proving object in the form of a passport page [36] and on the right, the output from the computer verifier for this passport page.

At the top left of the output, you can see what the page looks like under infrared light (marked "infrared"), and on the right, you can see what the same page looks like under ultraviolet light ("ultraviolet"). At the bottom of the column labeled "data page" is the holder's information read from the passport page using machine character recognition. In the adjacent column on the right is the passport holder's photograph read from the chip, and in the column labeled "chip" is the holder's information read from the chip. The column on the left shows the checks performed by the computer, which indicate that the passport being checked is genuine. The example above clearly shows that modern ID cards combine various security techniques to increase the credibility of personal identification (printed techmetrics, biometrics, and passkey).

The techmetrics described so far are based on the appearance of static structures. These structures include printed patterns, wrinkled surfaces of materials, and the distribution of reflective or magnetic particles. Techmetrics based on the activity of an object are currently mostly in the research stage, with the most promising being action techmetrics from the family of so-called physically unclonable functions (PUFs) [37]. The concept of unclonable functions is based on the fact that no object can

be manufactured with absolute precision. As a result, the values describing the activity of each object change in the form of one of many possible time or other courses. If this course is unique and stable during repeated measurements, it can be used to identify the object.

Probably the best-known application of this approach is the identification of radio devices (known as RF fingerprinting). Here, slight deviations in parameters such as oscillator drift and power amplifier non-linearity are specifically exploited. These deviations are within the tolerance range and therefore have no effect on the radio communication itself. On the other hand, the measured signals exhibit unique characteristics that can be used to identify devices and cannot be imitated. Figure 11 shows the time curves of the signal at the start of transmission for four different Wi-Fi network cards [38]. It is clear from these curves that the individual cards can be distinguished based on these curves.

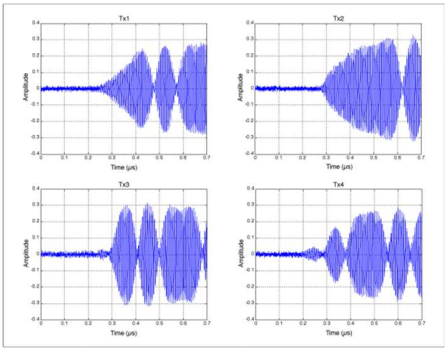

Fig. 11: Time course of the start of transmission of four Wi-Fi network cards [38]. The cards are labeled Tx1 to Tx4.

6. Verification by password

In verification by password, the proving factor p is secret data, and the carrier of this data is a person. This data must therefore be memorable in terms of its volume and variability. It most often takes the form of a string of alphanumeric characters, with the keyboard as the boundary module. There are also known proving factors in the form of graphic images (e.g., [39]), where the boundary module is either a touch screen or a computer monitor with a mouse. However, we will only focus on text passwords in the following, as graphic passwords have not yet become widespread. They are less secure (they can be relatively easily observed [40]) and, compared to a keyboard, require more expensive boundary modules.

For historical reasons, text passwords have various names (e.g., "Personal Identification Number" – PIN, passphrase, etc.), but we will use the uniform term "password" for them. The practical use of passwords has been documented since at least the 2nd century BC in the army of ancient Rome [41]. Night guards used them to determine whether another guard was approaching in the dark or whether it was potentially hostile persons. Verification by password is often used in contemporary electronic systems to control access to physical spaces (typically rooms or premises), but it is most commonly used to control access to data services provided, for example, by a user's personal computer, email server, etc.

In physical access control systems, a determination method is used. The proving factor for a person with identifier i_x is the password p_x, which most often takes the form of a string of 4 decimal digits and is called a PIN ("Personal Identification Number"). The verifier disposes of a verification list with pairs (i_x, v_x), where the verification factor $v_x = p_x$. The person does not present themselves during identification and simply enters their password using a keypad located in front of the entrance to the controlled area. By not providing their identifier, the person shortens the verification time and prevents queues from forming at the entrances. The keyboard converts the keystrokes into proving data $e = p_x$, which it passes to the test block T. The test block begins searching its verification list. When it finds a pair (i_x, v_x) for which $v_x = e$, the corresponding person is declared to be the person with identifier i_x.

In the case of determination, passwords must be assigned by the system administrator, and persons cannot choose their own passwords. The reason for this is that the verification factor must be unique in the determination method. Otherwise, situations would arise where the evidentiary data would match the verification data of several persons and identification would therefore not be unambiguous. In this context, the system administrator must also choose the number of digits in the PIN code appropriately, depending on the number of persons in the verification list. If l is the number of digits in the PIN, then the number of possible PIN codes is equal to $n = 10^l$. Then the probability P that the l-tuple of digits randomly entered by the attacker will be a valid PIN code (and the attacker will therefore gain access) is equal to $P = N/n$. For example, if a total of $N = 500$ persons have access to the building and $l = 4$, then $P = 500/10^4 = 0.05$. In this case, the attacker will be successful on average after $1/P = 20$ attempts.

Data service access control systems use an authentication method. The proving factor for a person with identifier i_x is again the password p_x, for which a string of at least 8 alphanumeric characters is currently recommended ([42], p. 13). The verifier disposes with a verification list containing pairs (i_x, v_x), whereby in the oldest systems, the verification and proving factors were identical, i.e. $v_x = p_x$ [43]. Using their computer keyboard, the person entered not only their password p_x but also their identifier $i = i_x$. The computer keyboard then sent the pair (i, e) to the test module T, where the evidentiary data e was the same as the proving factor p_x. After receiving the pair (i, e), the test block T found the pair $(i_x = i, v_x)$ in the verification list according to the value i and then verified whether $e = v_x$. If so, the person was confirmed as the person with identifier i_x. In the case of

authentication, persons can choose their own passwords, because in this case the verification factors of other persons are not tested.

Verification using a password is technically very simple, but it has three critical points. The first is the reliable storage of the password on the person's side, the second is the secure transmission of the password to the test module, and the third is the secure storage and verification of the password in the test module. The problem of reliable password storage on the person's side is caused by a natural characteristic of human memory, which is forgetfulness. Various techniques are therefore recommended for remembering and retrieving a sufficiently strong password (e.g., [44]), but unfortunately, these do not address the security requirement that a person should have different passwords for different services (e.g., for email, banking, etc.). Nowadays, when people commonly use dozens of services, this can only be solved in practice by external password storage, which is either specialized hardware or specialized software running on the person's device (e.g., on a computer). But then it is no longer just about verification by password, but about aggregated verification by password and passkey (see below).

The second problem is the secure transmission of the password to the test module. If there is a risk that the connection between the boundary module F and the test module T is being tapped, encryption must be used. For remote verification, where the boundary module and test module are connected via a computer network, the TLS (Transport Layer Security) cryptographic protocol is very often used for encryption. For local verification, such as when a user logs on to their computer, the password is not encrypted. It is assumed that any eavesdropping would be so costly for the attacker that it would not be worthwhile. However, this is no longer the case. For example, Figure 12 on the left shows an inexpensive USB adapter that an attacker inserts between the keyboard connector and the computer port.

Fig. 12: On the left is a USB keylogger with a Wi-Fi interface [45] and on the right is a thermal image of the keyboard immediately after entering the password [46].

Since the keyboard is often connected to the USB port on the back of the computer, the user may not be aware of the adapter's existence. The adapter transmits the bits from the keyboard to the computer, stores them in its memory, and, if necessary, sends them to the attacker's device via a built-in Wi-Fi interface. The attacker can thus obtain not only the password to log into the computer, but also the passwords to all services to which the user has logged in via the network.

The eavesdropping channel described above belongs to a category of channels that we call parasitic. These are channels that an attacker creates by modifying the user's device. In our specific case, this involved connecting an eavesdropping adapter to the user's computer. In addition to parasitic channels, an attacker can also use so-called compromising channels. These exploit the properties of certain processes associated with the functioning of the user's device. The attacker does not modify the user's device, but simply monitors and evaluates its behavior or emissions in a suitable manner. An example of a compromising channel is heat traces on the keys (see Fig. 12 on the right). The keys that the user pressed when entering their password have a higher temperature than the other keys and are therefore clearly visible on a thermal image of the keyboard. The attacker can thus find out which keys the user pressed when entering their password and therefore what characters the password consists of.

Another possible attack is password guessing. The attacker uses the boundary module to send various text strings to the test module and hopes that one of the attempts will result in the correct password being entered. However, this type of attack can be easily eliminated by temporarily or permanently blocking further communication between the test module and the boundary module after a certain number of unsuccessful password attempts, or by gradually increasing the interval for testing the evidentiary data.

The final problem is secure verification of passwords in the test module. The most significant issue here is that communication and software weaknesses allow an attacker to copy the verification list from the test module. If the verification factor was the password itself (i.e., $v_x = p_x$), the attacker would discover all passwords and could impersonate any person on the list. For this reason, hash functions are currently used to create verification factors. A hash function H is a so-called one-way function, i.e., from the input value a, the output value $b = \mathrm{H}(a)$ can be determined, but on the other hand, it is practically impossible to find the corresponding input value a for any value b. It follows that if the verification factor has the form $v_x = \mathrm{H}(p_x)$, an attacker cannot determine the password p_x from this value.

In the above case, the most suitable method for the attacker is to use a dictionary attack. This is based on the fact that, for reasons of memorability, the set of passwords cannot be too large. During the attack, the attacker uses a dictionary of commonly used passwords, which can contain up to millions of entries (e.g., [47]), and, if necessary, expands it using various techniques, such as substituting letters with numbers (e.g., [48]). From this dictionary, they take individual passwords a_j, calculate their hash $b_j = \mathrm{H}(a_j)$ and create a dictionary of hashes from the pairs obtained (a_j,

b_j). It then searches for a match $b_j = v_x$ in the verification list and in its list. If such a match is found, it determines the password of the person i_x, since the equation $b_j = v_x$ implies that $H(a_j) = H(p_x)$ and therefore $a_j = p_x$.

To reduce the effectiveness of the described attack, the verification list for each person is extended by a unique number u_x, called salt. The verification factor can then take the form $v_x = H(u_x \| p_x)$, where the symbol $\|$ represents the concatenation of sequences of symbols (in this case, the sequences u_x and p_x). The advantage of the described technique is that the attacker's list of hashes $b_j = H(u_x \| a_j)$ is no longer universally applicable to any list and any person. An attacker with this list can only attack the password of person i_x at a single verifier and must create a different dictionary of hashes to attack the password of another person or at another verifier. However, creating a dictionary of hashes is time-consuming and computationally intensive, so the potential gain from a successful attack may be significantly less than the cost of the attack. This fact has a deterrent effect on attackers.

The use of slow hash functions is another way to deter attackers. Slow hash functions (e.g., [49]) are hash functions for which the calculation time can be set so that the calculation of a single hash is acceptably fast, but the calculation of a hash for the entire dictionary would be too time-consuming.

Another possible attack on the verification list is the so-called brute force attack. This consists of the attacker gradually generating all possible variations of strings of length l characters from the set of permissible L characters, hashing each created string a, and comparing the resulting hash $b = H(a)$ with the verification factor v_x. If they match, the password p_x of the person with identifier i_x is the string a. Now suppose that passwords have the recommended length $l = 8$ characters from the set $L = 70$ characters (lowercase and uppercase letters of the English alphabet, digits, and 8 symbols). The number of all such passwords is equal to the value $N = L^l = 70^8 = 5.8 \cdot 10^{14}$. Current technologies allow even individuals to achieve a hashing speed of $R = 10^{11}$ hash/s.

For example, according to [50], the commonly available RTX 4090 graphics card can achieve a speed of $R = 164.0$ Ghash/s $= 1.64 \cdot 10^{11}$ hashes/s for the frequently used MD5 hash function. An attacker with this equipment is then able to try all the passwords we are considering in a time $T = N/R = 5.8 \cdot 10^{14}/(1.64 \cdot 10^{11}) = 3537$ [s] ≈ 1 hour. From the above calculation, it is clear that the resistance time of passwords with a length of $l = 8$ characters is already unacceptably short, and so it is currently proposed to set the minimum password length at 15 characters ([51], p. 13). For passwords with a length of $l = 15$ characters from a set of $L = 70$ characters and for $R = 1.64 \cdot 10^{11}$, the resistance time $T \approx 914 \cdot 10^6$ years is obtained. This resistance time is already sufficient.

7. Verification by passkey

In the case of verification using a passkey, the proving factor p is secret data which, unlike passwords, cannot be remembered by a person. In electronic identification systems, the object must be equipped with suitable hardware that serves as a memory and, if necessary, as a unit for processing this secret data. The above-mentioned hardware allows the identifier of a given object to be proven, and we will therefore refer to it as proving hardware. The proving hardware communicates with the verification system via a suitable data interface. On the verification system side, this interface also serves as the boundary module F.

Proving hardware can be classified as:

- memory storage
  - without controlled access,
  - with controlled access,
- hardware for cryptographic verification.

In memory storage, the proving factor p and the evidentiary data e are the same, i.e. $e = p$. Memory storage without controlled access typically includes cards designed to control access of persons to physical spaces. Historically older are cards with a memory chip (so-called proximity cards) and newer are cards with a chip in the form of a microcontroller (so-called smart cards). In both cases, the proving factor is the number stored in the card chip, whereby reading this number is not regulated in any way. This is a major weakness, as an attacker can also read the secret value and then either create a copy of the card with this secret value (a so-called clone) or create hardware that successfully mimics the behavior of the card (a so-called emulator). Devices that can emulate (e.g., [52] in Fig. 13, left) or clone (e.g., [53] in the same figure, right) the aforementioned proving cards are now commonly available on the market. For this reason, memory storage devices without controlled access are gradually being phased out.

Fig. 13: On the left, the Flipper Zero access card emulator [52] and on the right, the iCopy X100 access card replicator [53].

Password manager is a typical example of controlled-access memory storage. The owner stores their identifier i_X and password p_X (which here has the meaning of a passkey)

for each verifier g_X in this storage. Access to all this data is controlled based on knowledge of the master password p. The user must therefore remember this single password, which gives them access to the memory storage containing the triplets (g_X, i_X, p_X) needed to prove their identity to dozens of different verifiers. This is essentially an aggregated verification, where verification by passkey p_X implicitly includes verification by password p. A password manager is usually software that is integrated into the owner's hardware, such as a computer or smartphone (see Figure 14 on the left). However, password managers also exist in the form of specialized hardware (see the same figure on the right).

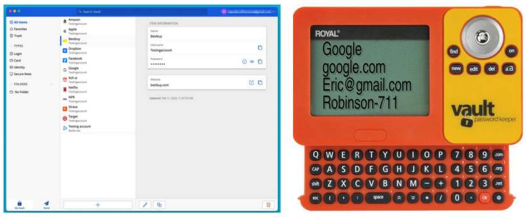

Fig. 14: On the left, an example of a software password manager [54]; on the right, an example of a hardware password manager [55].

The last and most promising type of proving hardware is hardware for cryptographic verification. This type is based on message encryption and authentication (see below) using both symmetric and asymmetric cryptosystems. In the case of symmetric cryptosystems, both communicating parties have a secret key k. The proving factor and the verification factor are then the same, i.e. $p = v = k$. In asymmetric cryptosystems, the keys on both sides are different. In proving hardware, a private key x is used, i.e. $p = x$, and the corresponding public key y is stored in the verifier, i.e. $v = y$. The private key is a secret key that is known only to the proving hardware, and the public key is a key that is not kept secret, so anyone can know it. A major advantage of cryptographic hardware is that the evidentiary data e is different for each verification. This prevents an attacker from using a so-called replay attack, where the attacker exploits evidentiary data captured in one of the previous verifications.

Cryptography ensures the security of messages using mathematical methods, so that messages, keys, and any other bit sequences are understood as numbers with which mathematical operations are performed. Within encryption ENC, message m is transformed using an encryption key into a pseudo-random number c, which is called a cryptogram c. If an attacker intercepts the cryptogram in the transmission channel, they are unable to determine what message is encrypted in it due to its pseudo-random nature. Symmetric encryption is formally written as $c = \mathrm{ENC}(m, k)$, which means that the message m is encrypted with a secret key k into form a cryptogram c. Asymmetric encryption uses a publicly known key y (i.e., anyone can perform encryption), and we will write encryption as $c = \mathrm{ENC}(m, y)$. In both types of cryptosystems, the cryptogram can be transformed into the original message only with the decryption key. The decryption function DEC is used for this transformation, where in symmetric cryptosystems the decryption key is a secret value k and in asymmetric cryptosystems it is a private key x. Formally, we will write this as $\mathrm{DEC}(c, k) = m$, or $\mathrm{DEC}(c, x) = m$. Examples of verification protocols based on symmetric ciphers are given, for example, in standard [56].

To ensure the authenticity of messages, message m is transmitted together with a pseudo-random number s, whose value depends on message m and on the secret key k or private key x. The authentication number s allows the recipient to verify that the message has not been altered during transmission and that the author of the message is hardware with knowledge of the secret k or private x. In symmetric cryptosystems, the number s is usually called the message authentication code (MAC), and the calculation of this number is formally written as $s = \mathrm{MAC}(m, k)$. In asymmetric cryptosystems, the authentication number s is called a digital signature, and the message signature is written as $s = \mathrm{SIG}(m, x)$. After receiving the pair (m, s), the recipient verifies the authenticity of the message as follows. In the case of a symmetric cryptosystem, the recipient calculates the own value of the authentication number $s' = \mathrm{MAC}(m, k)$. If $s = s'$, then the received message m is considered authentic. In the case of an asymmetric cryptosystem, a verification function $\mathrm{VER}(m, s, y)$ is defined for each digital signature algorithm, the output of which is a binary value w. If $w = 0$, then the message m is authentic, and if $w \neq 0$, then the received message was either altered during transmission or was not signed with the private key x. Examples of verification protocols based on authentication codes are given, for example, in standard [57], and protocols based on digital signatures can be found, for example, in [58].

In the case of hardware, the presentation time d of the entity is significantly shorter than the verification test time D, i.e., $d \ll D$. Therefore, only the authentication method is used for cryptographic verification. This is because verification in the case of authentication involves only the presentation of the proving hardware and a single test. Then, for the verification time $T_A = d + D$. In contrast, determination generally requires r tests, and usually $r > 1$. When using the determination method, the verification time $T_D = r \cdot D$, and for $r > 1$, it is clear that $T_D = r \cdot D > T_A = d + D$.

For scenarios where only one party is authenticated, there are basically three basic types of verification protocols, which are shown in Figure 15. On the far left is verification without a challenge, in the middle is verification with a challenge, and on the right is verification using a one-time password (OTP). The variable i is the hardware identifier, u is a unique number for a given verification (we will call it unique), and e is the evidentiary data. The unique u ensures

that the evidentiary data e has a different value in each verification. This prevents an attacker from using this data to deceive the verifier in future verifications.

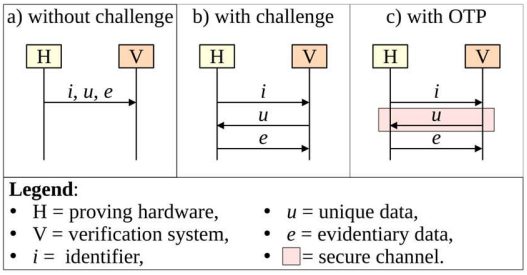

Fig. 15: Basic types of protocols for cryptographic verification. Figure a) shows the variant without a challenge, b) shows the variant with a challenge, and c) shows the variant with a one-time password (OTP).

First, we will describe verification without a challenge. In this case, both the hardware H and the verifier V must have synchronously functioning sources of unique values u. A frequently used source is the current time or a suitable counter (e.g., the verification sequence number). If the unique values change completely synchronously on both sides, the value u does not need to be transmitted. Otherwise, the unique must be transmitted because the test must be performed with the value u sent by the hardware. In this case, however, the verifier first checks whether the difference between the received unique and its own unique exceeds the specified tolerance limit. Depending on the cryptographic function used, the evidentiary data e takes the form of either a cryptogram $e = \mathrm{ENC}(u, k)$, an authentication code $e = \mathrm{MAC}(u, k)$, or a signature $e = \mathrm{SIG}(u, x)$. The hardware then sends the triplet (i, u, e) to the verifier.

The verifier finds the verification factor v based on the identifier i and uses it to test the evidentiary data e. In the case of a cryptogram, it decrypts it using $v = k$, where it must obtain the value u, i.e., $\mathrm{DEC}(e, k) = u$ must hold. For an authentication code, it calculates its own value $e' = \mathrm{MAC}(u, k)$. If the received e is equal to the calculated e', the test is successful. In the case of a signature, the received values u and e are substituted into the verification function and $\mathrm{VER}(u, e, y) = 0$ must be true.

An example of hardware for verification without a challenge is shown in Fig. 16. The device shown is used in Internet banking. Both the hardware and the verifier have synchronously running clocks, whose current time t changes every minute. The value t is used as a unique value u. The time stability of the clocks is so high that it is not necessary to transmit the unique. A MAC-type cryptographic function is used to calculate the evidentiary data, the output of which is converted to a six-digit decimal number. This is displayed on the hardware display as the bank client's evidentiary data e. The client logs in to the bank server and enters their identifier i and the currently displayed evidentiary data e on the login page. They send this information to the bank. As already mentioned, the unique $u = t$ does not need to be transmitted because both the hardware and the server know this value. The server uses the identifier i to obtain the verification factor $v = k$ and calculates its own six-digit number $e' = \mathrm{MAC}(t, v)$ for the current time value. If $e = e'$, then the logged-in client has hardware with the key k. And since this was assigned to the person with the identifier i, the client is the person i.

Fig. 16: RSA SecurID 700 evidentiary data generator [59].

The second type of protocol is verification with a challenge. In this case, the hardware and verifier do not have a synchronized source of unique values, and the unique values are generated unilaterally by the verifier. Unique can be, for example, random number, the current time of the verifier's clock, or payment transaction data. In the first step of the protocol (see Fig. 15), the hardware sends the identifier i. The verifier responds with the unique u, whereupon the hardware responds by sending the evidentiary data e. The calculation of the evidentiary data e and its testing in the verifier are the same as in the previous case. The disadvantage of the challenge protocol compared to the non-challenge protocol is slower verification, because three transmissions are required instead of a single transmission. On the other hand, the implementation of the challenge protocol is simpler, because the generation of unique values does not require any synchronization.

Fig. 17 shows an example of hardware for verification with a challenge.

Fig. 17: Example of hardware for verification with a challenge. On the left is the challenge sent in the form of a QR code with bank transaction details. On the right is the response (i.e., evidentiary data) for the transaction [60].

It is a smartphone with a suitable application that is used to authorize payment transactions. As part of this authorization, the client proves their identity and approves the transaction. When a transaction needs to be authorized

in internet banking, the bank's server sends a QR code with transaction data to the client's computer monitor. This code is the challenge u. The client photographs the QR code with their smartphone (image on the left) and the app then calculates the e evidentiary data (Response Code in the image on the right) from the proving factor and the challenge. The client enters this data into a form on the website on their computer and sends it to the bank's server. The bank's verifier checks the accuracy of the evidentiary data and then executes the transaction.

The last type of cryptographic verification protocol is verification using a one-time password (OTP). In the first step (see Fig. 15), the hardware sends the identifier i. The verifier then generates a one-time password *OTP*, which it encrypts using a symmetric or asymmetric cipher to form a unique value $u = \mathrm{ENC}(OTP, k)$ or $u = \mathrm{ENC}(OTP, y)$. This encryption creates a virtual secure channel in the physical transmission channel between hardware H and verifier V. This secure channel is shown in the figure as a red rectangle. The secure channel is shown in the figure as a red rectangle. The hardware decrypts the received unique value using its proving factor k or x, obtaining $\mathrm{DEC}(u, k) = OTP$ or $\mathrm{DEC}(u, x) = OTP$. The hardware then sends the decrypted value *OTP* as evidentiary data e, and the verifier checks whether this data matches the value of the *OTP* it generated. If $e = OTP$, the hardware has the proving factor k or x and the verification is successful.

The above solution requires that both the verifier and the hardware be capable of performing cryptographic operations. However, this entails higher costs, which is why in practice, one-time passwords are often transmitted in unencrypted form as $u = OTP$. However, for security reasons, the value u must be transmitted in a different physical transmission channel than the one used to transmit the identifier i and evidentiary data e. In Fig. 15, this secure channel is shown as a red rectangle. Channels that are assumed to be secure against an attacker with expected capabilities, i.e., an attacker will not be able to obtain the transmitted *OTP*, are used. In practice, this most often means transmitting the *OTP* via SMS message (see Fig. 18) or email. In both cases, these are channels that are reserved exclusively for the given user and provide the necessary level of confidentiality against eavesdropping by most attackers.

The above verification using two physical transmission channels is generally no longer direct verification using an passkey, but a specific form of witness verification, where our verification system V_0 relies on the testimony of another verification system V_1. The verification system V_1 is part of the system that controls access to messages u delivered via a secure channel. The type of verification is then, of course, determined by the type of verification used in system V_1. In the case of SMS messages, this is passkey verification again, as SMS messages are only sent to phones equipped with a SIM card for the given phone number. This SIM card is essentially a microcontroller that authenticates itself to the telephone network using a secret key. However, in the case of email messages, password authentication is most common.

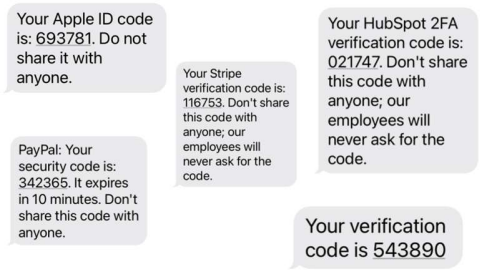

Fig. 18: Examples of one-time passwords sent in an SMS message [61].

8. Conclusion

Finally, a few remarks on the use of electronic identification systems in practice. Object identification systems usually identify only by presentation, i.e., the identifier is obtained based on what the object itself states about its identity (if it is capable of doing so) or based on what is stated on it. Examples of such systems include the identification of vehicles by their license plates, the identification of computers by the IP addresses of the sent packets, the identification of aircraft and ships by the data they transmit by radio, and the identification of goods by the barcode labels affixed to them.

Object identification systems requiring entity verification are most commonly found in the form of systems for authenticating servers and network elements in computer networks (passkey verification). Electronic systems for verifying personal identification cards and banknotes (technometric verification using printed security features) are also very widespread.

Electronic systems designed to identify persons have an exclusively verification character. Biometric verification, password verification, or verification by means of an object can be used, whereby verification by means of an object can be based on techmetrics or a passkey. Each type of verification has its advantages and disadvantages, and their application therefore depends on the specific scenario. However, given the current state of technology, the following trends have emerged.

If the system administrator does not have sufficient control over the boundary modules and their environment (typically in remote verification), attackers can modify these modules or attempt various forms of forgery. In addition, communication with boundary modules must be cryptographically protected to ensure secure transmission of evidentiary data. For such a scenario, verification using proving hardware is the most secure option. The disadvantage of this solution is that the person must have

sufficiently powerful hardware. However, this is not currently a problem, as the necessary hardware is widely available in everyday life (e.g., smartphones or tablets). To increase the security of the aforementioned solution, aggregated verification is gradually being implemented, whereby the proving hardware first authenticates its owner. Either biometrics or a password is used for this direct verification. Biometric verification is more expensive but more convenient for users, while password verification is cheaper but less convenient.

The concept described above is promoted by the FIDO (Fast IDentity Online) alliance [62]. According to the standards of this alliance, each person has proving hardware (e.g., a computer or smartphone) and uses it to verify their identity on servers using a digital signature. However, before this verification, the person must first authenticate themselves to their proving hardware.

In systems where both the border modules and transmission channels are secured and controlled by a single administrator, any verification option can be used. In systems where the person is in the position of a citizen of a certain country, verification using an ID card is widely used. Printed techmetrics and, where applicable, cryptographic verification are used to verify the authenticity of the ID card, and the verification factors of the person stated on the card are then used for biometric authentication of the cardholder. In systems where the person is in a position other than that of a citizen (typically access systems to buildings and premises), biometric or password verification is most commonly used for personal verification. As already mentioned, biometric verification tends to be more expensive but more convenient for persons, while password verification is the opposite. However, due to the widespread availability of portable computing devices (typically smartphones), cryptographic verification is also beginning to be used here.

References

[1] Shirey R.: *Internet Security Glossary, Version 2*. [RFC 4949]. Internet Engineering Task Force, Fremont 2007. Available at: https://datatracker.ietf.org/doc/html/rfc4949

[2] NXP Semiconductors: *MIFARE product and handling of UIDs*. Application note AN10927. Eindhoven 2018. Available at: https://www.nxp.com/docs/en/appliction-note/AN10927.pdf

[3] Merriam-Webster: *determination*. Encyclopædia Britannica. Springfield. Available at: https://www.merriam-webster.com/dictionary/determination

[4] Vacca J. R.: *Biometric Technologies and Verification Systems*. Butterworth-Heinemann, Burlington, 2007.

[5] Das R.: *Security Technology for Identity Verification*. Taylor & Francis, New York 2019.

[6] Merriam-Webster: *identifying*. Encyclopædia Britannica. Springfield. Available at: https://www.merriam-webster.com/dictionary/identifying

[7] NIST Glossary: *identification*. National Institute of Standards and Technology, Gaithersburg. Available at: https://csrc.nist.gov/glossary/term/identification

[8] Jain A., Hong L., Pankanti S.: Biometric identification. Communications of the ACM, Volume 43, Issue 2, pp. 90 – 98. Available at: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=0bcdbff633e46b14b2ffc13d5b26455e8e1ded71

[9] IDEMIA. *MorphoWave XP: Contactless fingerprint terminal with extended performance*. Available at: https://www.idemia.com/wp-content/uploads/2022/01/morphowave-xp-idemia-brochure-202201.pdf.

[10] LIPS: *LIPSFACE On-Device 3D Facial Recognition*. LIPS Corporation, Neihu. Available at: https://www.lips-hci.com/lipsface-3d-facial-recognition-solution

[11] IDEMIA: *Contactless fingerprint*. IDEMIA, Courbevoie. Available at: https://www.idemia.com/contactless-fingerprint

[12] IRISID: *IrisAccess 7000S*. Iris ID Systems, Cranbury. Available at: https://www.irisid.com/productssolutions/hardwareproducts/

[13] HFSECURITY: How biometric palm vein ensures effective workforce management. Huifan Technology, Shenzhen. Available at: https://hfsecurity.cn/biometric-palm-vein/

[14] Neak Media. *HVPR VP-II X Hand Vascular Scanner*. TechSphere, Riverside. Available at: https://hvpr.us/products/

[15] HITACHI. *USB Finger Vein Biometric Authentication Unit*. Hitachi, Tokio. Available at: https://www.hitachi.com/products/it/veinid/products/embedded_devices_u .html

[16] Clark M.: *Hand Geometry Recognition Biometrics: All You Need to Know*. Bayometric, Fremont. Available at: https://www.bayometric.com/hand-geometry-recognition-biometrics/

[17] Descartes Biometrics: *HELIX. Lock Screen App*. Descartes Biometrics, Blaine. Available at: https://www.descartesbiometrics.com/helix-app/

[18] Zhou R., Zhong D., Han, J.: *Fingerprint Identification Using SIFT-Based Minutia Descriptors and Improved All Descriptor-Pair Matching*. Sensors, 13(3), 3142-3156. Available at: https://www.mdpi.com/1424-8220/13/3/3142

[19] Hazarika P., Russell, D.A.: *Advances in fingerprint analysis*. Angewandte Chemie, 51 15, 3524-3531. Available at: https://www.semanticscholar.org/paper/Advances-in-fingerprint-analysis.-Hazarika-Russell/dd232f23958e5b2782b64771305827ea76f5284c

[20] WATRIX: *Gait capture array*. Galaxy Water Drop Technology, Jiangsu. Available at: http://watrix.ai/product/foresight/?productType=3

[21] GNANI: *Armour365 Voice Biometrics*. Gnani.ai, Bengaluru. Available at: https://voicebiometrics.ai/

[22] ARATEK: *How behavioural biometrics are improving security*. SourceSecurity.com, London. Available at: https://www.sourcesecurity.com/news/behavioral-biometrics-improving-security-discusses-aratek-co-1669082358-ga.1671195243.html

[23] Engelsma J. J. et al.: *Universal 3D Wearable Fingerprint Targets: Advancing Fingerprint Reader Evaluations*. In IEEE Transactions on Information Forensics and Security, vol. 13, no. 6, pp. 1564-1578, June 2018. Available at: https://arxiv.org/pdf/1705.07972

[24] Reinhard J. et al.: *Automatic data-driven design and 3D printing of custom ocular prostheses*. Nature Communications 15, Article number 1360 (2024). Available at: https://www.nature.com/articles/s41467-024-45345-5

[25] TOXEL: *Human Face Masks*. Toxel.com. Available at: https://www.toxel.com/tech/2011/10/15/human-face-masks/

[26] Schuiki J., Prommegger B., Uhl A.: *Confronting a Variety of Finger Vein Recognition Algorithms With Wax Presentation Attack Artefacts*. 2021 IEEE International Workshop on Biometrics and Forensics (IWBF), Rome, Italy, 2021, pp. 1-6. Available at: https://ieeexplore.ieee.org/document/9465091

[27] CBA: *Voice Biometric Authentication in Call Centers: How to Fight Spoofing*. Communication Business Avenue. Available at: https://cba-gbl.com/voice-biometric-authentication/

[28] Tolk, K.: *Reflective particle technology for identification of critical components*. Tech. Rep. SAND-92-1676C, Sandia National Labs, Albuquerque, NM (1992). Available at: https://www.osti.gov/servlets/purl/7116334

[29] Buchanan, J., Cowburn, R., Jausovec, AV. et al.: *Fingerprinting' documents and packaging*. Nature 436, 475 (2005). https://www.researchgate.net/publication/7697181_Forgery_%27Fingerprinting%27_documents_and_packaging

[30] Cowburn R., Buchanan J.: *Optimisation*, US patent 8892556, Nov. 18. 2014. p. 7. https://patents.google.com/patent/US8892556B2/pt-PT

[31] MAGENSA: *Magensa Global MagnePrint Exchange. Card authentication for magnetic stripe cards in existence already*. MagTek 2019. Available at: https://www.magtek.com/content/documentationfiles/d99875243.pdf

[32] European Union Intellectual Property Office: *Anti-counterfeiting technology guide*. Alicante, Spain: The European Observatory on Infringements of Intellectual Property Rights, 2021. ISBN 978-92-9156-286-6. Available at: https://op.europa.eu/en/publication-detail/-/publication/a346f801-8d1c-11eb-b85c-01aa75ed71a1/language-en

[33] *Ausschnitt einer Guilloche auf einem alten 5-DM-Schein*. Online. In: WIKIPEDIA. Available at: https://commons.wikimedia.org/wiki/File:Guilloche.jpg

[34] *Protective elements – CZK 1000*. Online. In: CNB. Czech National Bank. Available at: https://www.cnb.cz/en/banknotes-and-coins/banknotes/protective-elements-czk-1000/

[35] Kluge, F.: *Erkennung der Ovi-Technik auf einer 50 Euro Banknote*. Online. In: WIKIPEDIA. Wikipedia. Available at: https://commons.wikimedia.org/wiki/File :Ovi.png

[36] *Datenseite des deutschen Reisepasses von Erika Mustermann*. Online. In: WIKIPEDIA. Wikipedia. Available at: https://de.wiktionary.org/wiki/Erika_Mustermann

[37] Maes R.,Verbauwhede, I.: *Physically Unclonable Functions: A Study on the State of the Art and Future Research Directions*. In book: Towards Hardware-Intrinsic Security. Springer, Berlin, October 2010. Available at: https://www.researchgate.net/publication/226371108_Physically_Unclonable_Functions_A_Study_on_the_State_of_the_Art_and_Future_Research_Directions

[38] Ureten O., Serinken N.: *Wireless security through RF fingerprinting*. In Canadian Journal of Electrical and Computer Engineering, vol. 32, no. 1, pp. 27-33, Winter 2007. doi: 10.1109/CJECE.2007.364330

[39] Biddle R., Chiasson S.: *Graphical passwords: Learning from the first twelve years*. ACM Computing Surveys (CSUR), Volume 44, Issue 4, Article No. 19, Pages 1 – 41. Available at: https://hotsoft.carleton.ca/~sonia/content/Chiasson_TR-11-01.pdf

[40] Aviv A. J. et al.: *Towards Baselines for Shoulder Surfing on Mobile Authentication*. ACSAC '17: Proceedings of the 33rd Annual Computer Security Applications Conference, pages 486 – 498. Available at: https://www.researchgate.net/publication/319875717_Towards_Baselines_for_Shoulder_Surfing_on_Mobile_Authentication

[41] Polybius: *Histories*. Book 6, Daily Orders and Watchwords. Available at: https://www.perseus.tufts.edu/hopper/text?doc=Perseus%3Atext%3A1999.01.0234 %3Abook%3D6%3Achapter%3D34

[42] Grassi P. A. et al.: *Digital Identity Guidelines. Authentication and Lifecycle Management*. [SP 800-63B]. National Institute of Standards and Technology, Gaithersburg 2017. Available at: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63b.pdf

[43] Morris, R., Thompson K.: *Password Security: A Case History*. Bell Laboratories. N. Jersey 1979. Available at: https://rist.tech.cornell.edu/6431papers/Morris Thompson1979.pdf

[44] Rubenking N. J.: *3 Simple Tricks for Remembering Strong Passwords*. PCMag, Nov. 11. 2024. Available at: https://www.pcmag.com/how-to/tricks-for-remembering-strong-passwords

[45] VENTURE VISION: *AirDrive Keylogger Pro*. Venture Vision, Miami. Available at: http://www.airdrive-wifi.com/?page=AD020KLOGPRO

[46] SCHOOL OF COMPUTING SCIENCE: *Thermal Imaging Attacks*. University of Glasgow. Available at: https://www.gla.ac.uk/schools/computing/research/researchsections/gist-section/thermalimagingattacks/

[47] Cannon D.: *Crack more passwords with custom wordlists*. North Green Security. Aug. 21. 2024. Available at: https://northgreensecurity.com/2024/08/21/crack-more-passwords-with-custom-wordlists/

[48] Bobby P.: *Password Cracking using Focused Dictionaries*. SANS Institute. Available at: https://www.giac.org/paper/gsec/42/password-cracking-focused-dictionaries/100346

[49] Biryukov A. et al.: *Argon2 Memory-Hard Function for Password Hashing and Proof-of-Work Applications*. [RFC 9106]. Internet Research Task Force. Sept. 2021. Available at: https://datatracker.ietf.org/doc/rfc9106/

[50] Chick3nman: *RTX_4090_v6.2.6.Benchmark*. GitHub Gist. 2022. Available at: https://gist.github.com/Chick3nman

[51] Temoshok D. et al.: *Digital Identity Guidelines. Authentication and Authenticator Management*. [Draft SP 800-63B-4 2pd]. National Institute of Standards and Technology, Gaithersburg 2024. Available at: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63B-4.2pd.pdf

[52] FLIPPER: *Flipper Zero. Multi-tool Device for Geeks*. Flipper Devices, Claymont. Available at: https://flipperzero.one/

[53] AMAZON: *NFC RFID Card Copier Reader Writers*. Amazon.com, Seattle. Available at: https://www.amazon.co.uk/Copier-English-Version-Keyfobs-13-56MHz/dp/B0BF9Y9F45

[54] Eddy M.: *The Best Password Managers*. Wirecutter, Feb. 28. 2025. Available at: https://www.nytimes.com/wirecutter/reviews/best-password-managers/

[55] WALMART: *Digital Password Vault*. Walmart.com. Available at: https://www.walmart.com/ip/Royal-39226U-PV1-Digital-Password-Vault/39802008

[56] ISO: *IT Security techniques — Entity authentication, Part 2: Mechanisms using authenticated encryption*. [ISO/IEC 9798-2:2019]. Geneva, 2019.

[57] ISO: *IT Security techniques — Entity authentication, Part 4: Mechanisms using a cryptographic check function.* [ISO/IEC 9798-4:1999]. Geneva, 1999.

[58] ISO: *IT Security techniques — Entity authentication, Part 3: Mechanisms using digital signature techniques*. [ISO/IEC 9798-3:2019]. Geneva, 2019.

[59] RSA: RSA SECURID AUTHENTICATORS. EMC Corporation. Hopkinton, 2011. Available at: https://www.tokenguard.com/datasheets/2305_h9061-sid-ds-0212.pdf

[60] SOLIDPASS: Transaction Data Signing (TDS) Security Token. SolidPass. Available at: http://www.solidpass.com/authentication-methods/transaction-data-signing-tds.html

[61] Cazalet H.: What is SMS OTP? A simple guide for 2023. The SMS Works. Bristol, 2023. Available at: https://thesmsworks.co.uk/blog/SMS-OTP/

[62] FIDO: *How Passkeys Work*. Fido Alliance. Mountain View. Available at: https://www.passkeycentral.org/introduction-to-passkeys/how-passkeys-work

Karel Burda received the M.S. and Ph.D. degrees in Electrical Engineering from the Liptovsky Mikulas Military Academy in 1981 and 1988, respectively. During 1988-2004, he was a lecturer in two military academies. At present, he works at Brno University of Technology. His current research interests include the security of information systems and cryptology.