sciendo INTERNATIONAL JOURNAL ON SMART SENSING AND INTELLIGENT SYSTEMS

Research article | DOI: 10.2478/ijssis-2024-0035 Issue 1 | Vol. 17 (2024)

Extended Authentication Based on Geometric Patterns and Transformations

Tomasz Krokosz* and Jarogniew Rykowski*

Department of Information Technology, Poznań University of Economics and Business, Poland

*E-mails: tomasz.krokosz@ue.poznan.pl, rykowski@kti.ue.poznan.pl

Received for publication March 02, 2022.

Abstract

Authentication by key is successfully applied in many cases. However, this simple approach raises several problems during contextual authentication, when the final decision depends not only on the knowledge of the key but also on some other independent parameters such as time, distance, values fetched from nearby internet of things (IoT) devices, etc. Such contextual parameters are usually considered an independent phase of the whole authentication process. In this paper, we propose a new method of extended authentication because, instead of using just keys, we also applied some mathematical formulas describing the context to be evaluated as a single cryptographic process.

Keywords

master key, temporary key, onetime key, aging key, key verification, authentication, contextual authentication, security

I. Introduction

The target and final form of the authentication method follow from organizational and communication possibilities. For example, concerning paper documents, handwritten signatures, stamps, or notarial certificates are used. For electronic documents, this can be, e.g., a digital signature [1] or message authentication code (MAC [2]). In the case of citizen verification, authentication may be performed using biometric security [3], passwords, tokens, or ID documents. In the case of communication among nodes of a computer network, some methods based on onetime passwords [4], zero-knowledge proof [5], the symmetric key (symmetric cryptography [6]), or the private and public keys (as well as their pairs) as part of asymmetric cryptography [7] are applied. The last two methods constitute the classic key-based authentication scheme.

Symmetric encryption algorithms enable plaintext encryption and decryption of the ciphertext using one common key. The same parameter value (called a key) is the basis for encryption and decryption methods. Therefore, it may be carried out in the same way by each party of the communication process (Figure 1). The security of symmetric algorithms is related to the difficulty of finding the private key, which results from the length or the complexity of the processing methods. The second important issue is safe and effective key distribution, i.e., handing over the key to all parties of the communication in a way that does not lead to its disgrace. Existing and currently used symmetric encryption schemes are based on block ciphers mechanisms (e.g., ECB, CBC, CFB [8]) and stream cipher [9].

Symmetric encryption property is as follows:

$$D_K[E_K[M]] = M \tag{1}$$

With symmetric encryption, excluding the previously mentioned issues, e.g., key distribution, two problems are raised related to authenticity and

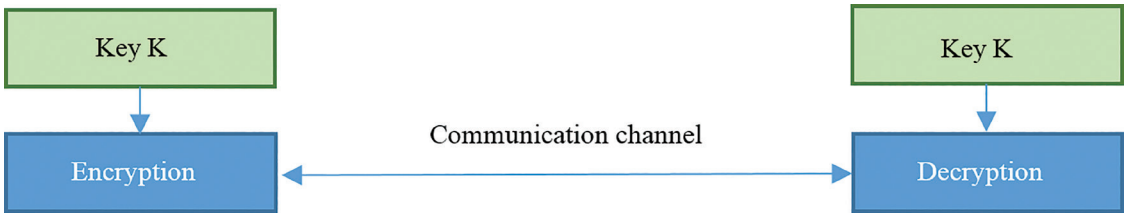

Figure 1: Communication in a symmetric system.

scalability. In the first case, since the parties share a key, it is not possible to prove (without additional measures) which of them is the actual sender. The communication channel is safe until the intruder becomes the owner of the key. In terms of scalability, there is a need for each party to store the keys of the other parties. In a system of four users, each user must remember six keys. For 10 users, the number of keys to remember increases to 45. However, their number may be ultimately reduced using a different way. A group of users can agree to use the same shared key. Then, each member of the group is required to remember and use only one key—to encrypt and decrypt data. Of course, such an agreement requires an increased level of trust in relation to sharing keys within individual pairs—in the case of a compromising key, not only it remains unknown which group member is responsible for it but also the detection of this fact requires several additional verification activities with respect to all group members.

The ciphers that use two different keys in operation (one for encryption and the other for the decryption of the message) belong to a group of asymmetric ciphers. One of the keys—the public key—is shared and used by interested parties in the data encryption process. The owner of the second key of the pair—the private one—can successfully convert the ciphertext to a plaintext (Figure 2). Thus, the essence of asymmetric encryption is the use of two interconnected keys, each of them has different applications. A user who wants to send an encrypted message to the recipient must encrypt the message with his (recipient) public key. After receiving the data, the receiver will use the private key, which is related to the public key used for encryption, to obtain an explicit message.

Asymmetric encryption property is as follows:

$$E_{K(public)}[M] = S. \tag{2}$$

While obtaining the plaintext from the ciphertext is performed as follows:

$$D_{K(private)}[S] = M. \tag{3}$$

Having two keys can provide the basis for a digital signature creation scenario. Person *A* has two keys (private and public) and the document that he wants to sign.

To perform the signature, person *A* encrypts the document using his private key and attaches the obtained ciphertext to the original document. Person *B* receiving a document allegedly signed by user *A* must be in possession of his public key. With it, he will decrypt the document, obtaining its open version. In the next step, person *B* verifies the obtained data. This provides certainty as to the correctness of the key—had the key of another user, the data could not have been decrypted with the public key of person *A*. It is also certain that the data content has not been modified. In the presented example, however, there are some nontrivial limitations. For example, a user receiving another user's public key does not have sufficient knowledge of it, e.g., does not know who the owner is. It also does not know if the received data actually represents a public key. For this purpose, the concept of certificates was developed. A certificate can be understood as an electronic attestation containing the necessary data, e.g., the public key and the aforementioned distinguished names, i.e., owner attributes, such as a name, and address, a serial number, etc. For this solution, the presence of a trusted third party is necessary, namely, an issuer of the certificate. Ultimately, the verification of the key and the author consists in verifying the certificate describing it.

Common and typical use of certificates (X.509) [10] covers, among others, signing documents, user authentication, or authentication and ensuring the establishment of a secure connection (SSL/TLS) [11]. It consists of, among others, the fields marked with the following labels: a version, a serial number, an issuer of the certificate, the validity, the entity for which it was issued, a public key, and the signature of the issuing authority. However, a verification of the certificate involves tracing the entire path of the trust chain in the public key infrastructure (PKI). As PKI provides authentication services, it becomes possible to authenticate entities and data, as well as confirm their

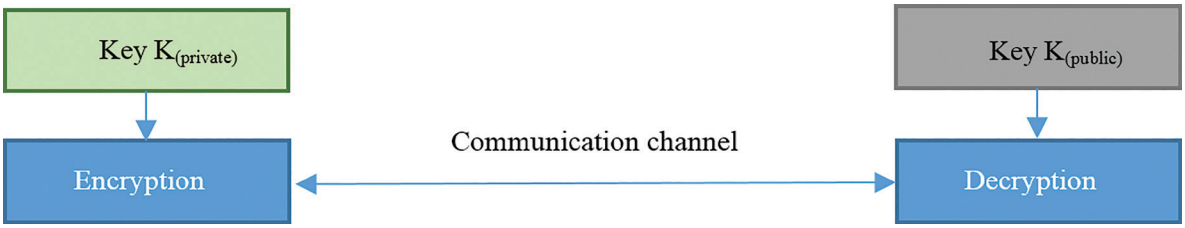

Figure 2: Communication in an asymmetric system.

integrity and nonrepudiation. PKI also guarantees confidentiality services, i.e., data protection against unauthorized disclosure and privacy.

Apart from the application, it is not without significance that one of the fields of the certificate is related to a public key. In turn, the validity period of this key is recorded in another field. In a matter of fact, these are two independent data—the fact of having the correct key does not indicate whether it is allowed to perform encryption processes by the organization issuing the certificate. Further, the validity of the key does not appear from the expiry date. Therefore, apart from checking the correctness of the encryption process, a series of operations must be performed to validate the remaining certificate data. Moreover, the validity of the key and the expiry date do not yet imply the correct use of these data, which depends on the context. For example, if a certificate is used to control access to an office, it should only be considered valid during office hours and invalid outside of this period of the day. At night, or on Sundays and holidays, this certificate should be rejected. In the case of control systems, the current state of such a system should determine the correct use of the certificate. For example, the user has the right to open the door only when he is waiting in front of it (the distance stands as a context), the right to turn on the lighting when it is dark (the brightness as a context), the right to enter when someone accompanies him (e.g., the context of a child accompanying by an adult), etc. As one can see, in a general case, it is impossible to predict all such metadata and save all of them in a single certificate.

The goal of this article is to propose a new method of verification using a PKI that uses a mathematical function instead of the metadata of the certificate for authentication. In our solution, we use functions that define certain geometric objects. For example, all the resources in the infrastructure of a company are represented in the form of circles—devices, rooms, actions to be performed, and users. If the user's circle is externally tangent to some other circle representing any resource, there is a relationship between the user and resource circles, i.e., the user has the rights to interact with it. Replacing the circles with circumferences also allows to verify variable values, e.g., elapsed time or distance. Such a distance or time may be represented as some points drawn on a line from the center of the circle to the tangent point (i.e., inside the circle described by the circumference). Verification of a point's location on a straight line allows for a quick determination of whether such a point lies inside the circumference (and thus the correctness condition is met) or outside. At the same time, this may be mathematically related to the change of a certain value, e.g., with the passage of time—if this value is outside the specified range at the time of verification, the verification will not be successful.

The above geometrical representation is even much more generic and is not related to only the simple relation of two entities (such as a user and a room). It may be noted that one circle may be tanged to several other circles and lines. Thus, the idea may be broadened to the solution that every entity with its geometric interpretation is related to many other entities, such as a single user has the same key to several rooms. This concept is related to so-called "master-key" technique, a widely used pure mechanical solution. This is an extremely useful scheme that allows access to all resources in a system with a single key. According to the idea, there is one key that allows opening all doors/locks within, e.g., a single building. Main key derivatives can be divided into separate groups, creating sets of group keys intended for, e.g., managers, cleaning personnel, or members of particular departments. Next, derivatives of group keys form a set of individual keys intended for the employees of the company.

Further structure of the text is as follows. The second section presents the basics of the master-key technique, widely used in classic access control. The third section is devoted to the problem of the mathematical determination of the context similarity. The fourth section deals with the possibility of uniform

implementation of methods described in the previous sections by means of circles and line equations. The fifth section presents the possibility of generalizing the solution and actually applying any mathematical function. As a result, there is no need to limit to a linear function and operate on straight line equations. In penultimate section, we focus on the literature review in relation to master-key technique. The solution summary and conclusions are included in the last part, which has been enriched with application examples and a description of possible application areas.

II Master-Key Technique

The principle of operation of the master key is as follows: in a situation where there are two locks and two keys, one of them can be used to open both locks, while the other only to open one of them. In the construction of locks, distinguish, among others, pins, driver-pins, or springs under each of them. In the case of an "ordinary" lock, i.e., a lock that can be opened with one key, if it is in a position that prevents its rotation (Figure 3A), it means that, e.g., the key used is incorrect.

Figure 3B shows the case where the lock can be opened with more than one key. As before, the illustration refers to one pin as the principle of operation. However, a number of pins applied is a choice of a designer and operator of the system.

By pushing the pin and making it placed at the correct position, one enables the rotation of the insert. It becomes possible only after entering the key that will properly arrange all pins in the lock. In Figure 3B, we can see that there are two positions where the insert will be no longer blocked. This makes it possible to prepare two separate keys for one lock, the selection of which will have the same effect—unlocking the insert and the possibility of turning the lock, i.e., opening the doors.

The master-key system provides access to all locks with a single key. The same key can effectively open a gate, door, drawer, or padlock—any available lock in the infrastructure of, e.g., a company. This significantly affects the convenience because people using the facility are exempt from the obligation to remember to carry a set of keys. Instead of a keychain, the user uses one key that gives him access to all or selected (related to the group he belongs to) locks. The master-key system reduces the number of circulating keys to a minimum. This results in further benefits related to facilitating their control and protection against unauthorized duplication.

Described techniques are successfully used in the following areas: single-family houses, companies, housing constructions (housing association), office buildings, public utility buildings, and shopping centers. The solution will also work in hotels, dormitories, kindergartens, schools, universities, and production facilities. Let's take the case of a hotel as an example. A guest booking a hotel room has access only to the rented room (and standard utility rooms, e.g., a staircase). His key/card is only able to open the door to which he has access. The cleaning-service organization can be carried out in a different way. Let us assume that a different cleaning person is responsible for each floor. Each of these people has their own key, with which they are able to open each door on the assigned floor and a dedicated storage room for the necessary utensils. This is ensured by entering group keys. Widening this example, other groups may include reception, technical, kitchen, and administration employees. Employees belonging to any group with their keys can only open the doors that are accessed by those belonging to the group. In the hierarchy of keys, directly above them, there is a general key that allows opening every door in the building. Described sample structure of the organization of keys in the master-key system is presented in Figure 4.

The solution presented under the name of master key is also sometimes marked with a different label, e.g., as a central opening system, a cap system, or a mother key. Regardless of the nomenclature, the final effect achieved is the same—using one key, users

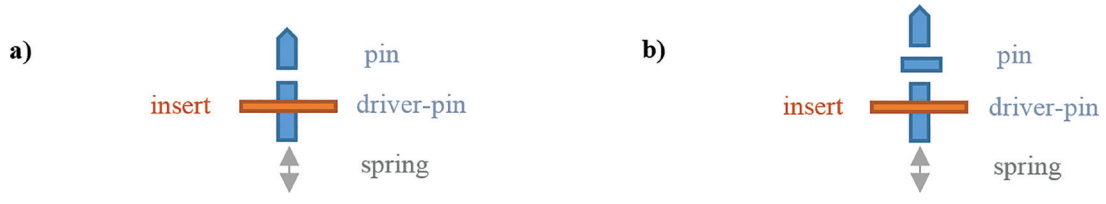

Figure 3: Examples of pin organization in a mechanical lock a) with one opening key b) with two opening keys.

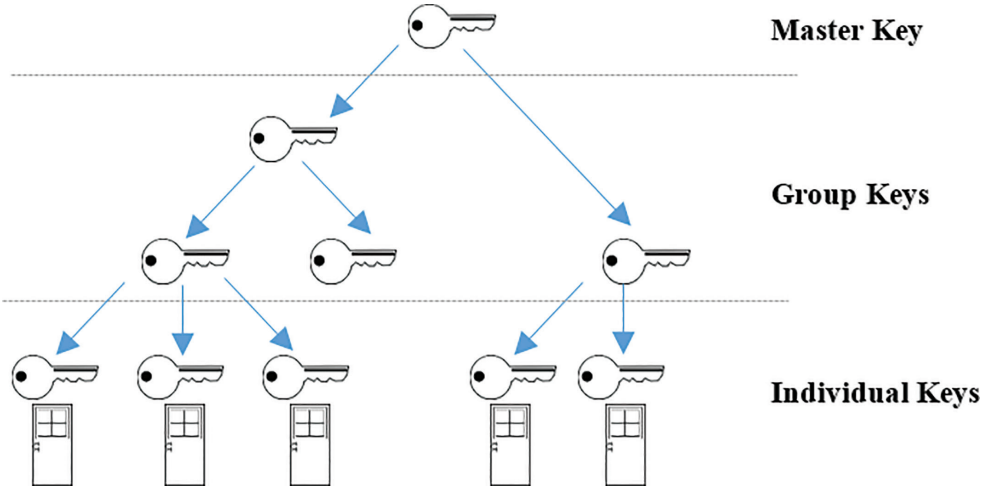

Figure 4: Hierarchy of keys in a master-key system.

can open all doors in a given facility, or only those to which they have access (group keys).

Note that a single lock can be opened with a single key. However, the same key may be used to open many locks. Returning to the geometrical examples we introduced in the previous sections, if one compares such a lock to a mathematical function, then the equation is such a unique function $y = f(x) = ax + b$. Each function value (y) has one and only one parameter (x) for which it can appear in the solution. In order to increase the number of keys, we can use the ambiguity of assigning parameters to the result. For example, for a quadratic function ($y = f(x) = ax^2 + bx + c$) each function result will be determined by two parameter values ("positive" and "negative" x values). As mentioned earlier, for the equation of a circle, "correct" references can exist to all points that lie on a straight line from the center of this circumference to the point-of-tangent with another—there are infinitely many such points, but at the same time, they are all unambiguously verified in terms of "correctness."

III Mathematical Determination of the Context Similarity

As a first approximation to present our solution, let us choose the circle equation described. It is an unambiguous function—either the point is inside (including the circumference itself) or outside. Assuming that the key (i.e., the coordinates of a point on the plane) is valid only inside the circumference, one obtains a verification function for which infinitely many values of the key are (potentially) correct. By properly manipulating the diameter of the circle and the location of its center, one can influence the probability of choosing a single value from the "correct" key values. For an infinitely small circumference, the probability is also infinitely small, but we still allow infinitely many "correct solutions."

Thus, our solution is based on the assumption that every resource in the system, including users, is represented on the plane with a single circle of a specific center point and diameter. If the circles of two resources are externally tangent, these resources are given mutual "access" to each other. Formally, it is when the distance between the centers of the circles $|S_1S_2|$ is equal to the sum of their rays, e.g., $d = r_1 + r_2$ (Figure 5).

Between the center of the circle and the tangent point, one may define a straight line. The equation of this line allows entering variable (or "fuzzy") keys. Once one associates some external and independent data with the plane on the straight line, e.g., the passage of time, distance, the aforementioned brightness, humidity, etc., the line (and its mathematical formula) may represent the variations of such data. Assuming that the correct value is any point lying on this straight line inside a given circumference, one obtains a set of valid keys of infinite cardinality. If the verification point "runs" in a straight line, changing its position on the basis of some external rules (e.g., a straight line symbolizes the passage of time), one can generally speak of "aging" keys that lose their "validity" after some time, at a larger distance, etc.

From a mathematical point of view, in our solution, each system element (user/resource or any other entity) is represented as a circle. Each circle is described by two coordinates, which define the position of its center and a variable with its radius. User has access to the resource when two circles have one and only one common point, i.e., they are externally tangent. For two circles $O_1 = (1; -2)$, $r_1 = 3$ (center of circle at the point (1; –2), radius of length 3), $O_2 = (-4; -2)$, $r_2 = 2$ (center of circle at the point (–4; –2), radius of length 2), one determines their equations from the formula in canonical form:

$$(x-X_1)^2 + (y-Y_1)^2 = r^2. \quad (4)$$

The values of X_1, Y_1 are the center points of the circle, and r is the radius of the circle. For values given earlier, the equations of the circles will be as follows: $O_1 = (x-1)^2 + (y+2)^2 = 9$ and $O_2 = (x+4)^2 + (y+2)^2 = 4$. Then it is necessary to calculate the sum of the radii, i.e. $r_1 + r_2 = 3 + 2 = 5$. Next, the value to be calculated is the absolute value from the difference of radii, i.e., $|r_1 - r_2| = |3-2| = 1$ (subtraction in combination with the absolute value is commutative, so the order of the elements does not matter).

The last parameter is the distance between the centers of the circles, calculated from the formula:

$$|O_1O_2| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}. \quad (5)$$

After substituting the earlier values into formula, the obtained value is 5 ($\sqrt{(-4-1)^2 + (-2-(-2))^2} = \sqrt{25}$).

For the above example, the function that checks if the sum of the radii of the circles is equal to the distance between the circles (outer tangency condition) would return the logical value *true* (substantially 5 = 5).

An important issue in the solution is ensuring access for a specified period, e.g., respecting the hotel's day and night access to the room only during the booking hours. The goal was achieved as follows. Each object representing a specific resource in the infrastructure, e.g., of a company, has specific coordinates: circle determines its equation, straight—linear function: $f(x) = ax + b$, parabola—quadratic function: $f(x) = ax^2 + bx + c$, etc. Mentioned examples of objects can be successfully applied to the coordinate system, which is their graphical representation. The periodic determination of the access time was achieved by linking the coordinates of the point (center of the circle) with the date and time for said reservation—being the beginning of the validity of the rights. Regarding the example, let us assume that the resource is accessed from 12/1/2020, at 1:15 PM. Then the center of the circle will have the following coordinates: (20201201, 1315). Then, in order to link the permission to the resource, it is needed to establish a contact point with the latter, i.e., make externally tangent circles. In this way, association of the resource (the room) with the user takes place. A straight line is drawn between the center of the circle and the point-of-tangent with the second circle. Checking whether the user has access to the resource at a certain time is carried out by verifying that the point (current date and time) belongs to this line. Consider two examples: in the first, the point will belong to the line (access it to be granted), and in the second, it will not be (the access is denied).

So far, for the purposes of the example, we used a simple linear function to precisely present the method of necessary calculations and their verification. With a

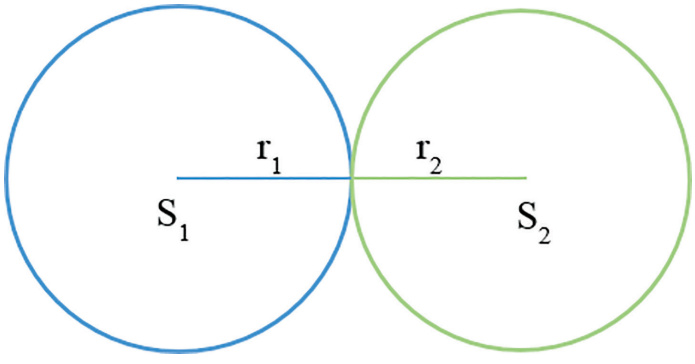

Figure 5: Externally tangent circles.

linear function with the equation: $f(x) = x + 3$, checking whether point P (1, 4) belongs to a straight line comes down to substituting values in the following way: $4 = 1 + 3$. i.e., we check whether the value of the second coordinate of the point P: 4 is equal to the sum of the first coordinate of the point P: 1 and the constant term in the linear function: 3. At the point Q (2, 3), we check if $3 = 2 + 3$. In this case, there is no equality, which means that the point does not depend on the line. This is related to the lack of access to the resource—e.g., the authorization period has not started yet, has already expired, or detected lack of resources associated with the user.

IV Specifying the Context in Combination with the Master-Key Technique

This section presents the master technique in which, using one key, it is possible to gain access to a whole group of rooms. It is permissible to designate group keys from the master key. In turn, the derivatives of group keys are individual keys. The graphical form of the solution is shown in Figure 6.

The topic can be associated with the following example scenario. Company's director has access to the entire infrastructure (master key). He can open all doors, perform any action, realize all operations. Next are the heads of departments/units who belong to the group of users defined by their position and have a group key. In their case, they can implement the access that their subordinates are entitled to (individual keys) and additionally have those dedicated only to them. The path of the recess can be much longer and need not be limited to just three levels.

As can be seen, all group and individual keys are derived from the master key. The latter is the basis, which is *de facto* an initialization variable for the generator of all subsequent related keys. For this reason, in the key generation algorithm, the "inheritance" method should be used. It is enough in the loop, during iterating over each newly added object (a derivative key), to bind it to the initialization value (master key), or a previous key, or even any key from the group. The matter is not only about the type of function and the number of arguments but also includes the domain—that is, the set for which the function exists.

Returning to the previous section in which we discussed the "aging" of the keys, we may now combine the "master-key" technique and "aging" keys using the same mathematical representation. For the latter, identifying the resources (users, achievable goods) in the form of circles, and assuming that access to the resource is achieved when the circles are externally tangent, was needed as a way to determine the limit of access validity. For this purpose, the coordinates of the center of the circle (user) have been linked with the tangent point with the accessed resource (Figure 7). The equation for the straight line from the center of the user's circle to the tangent point is equivalent to the "aging" key. Each submitted request is submitted at a specific time. As a result of linking the center of the circle and the tangent with the resource (their coordinates) with time, such verification becomes possible.

We can effectively combine the implementation of the "master-key" technique and the technique of

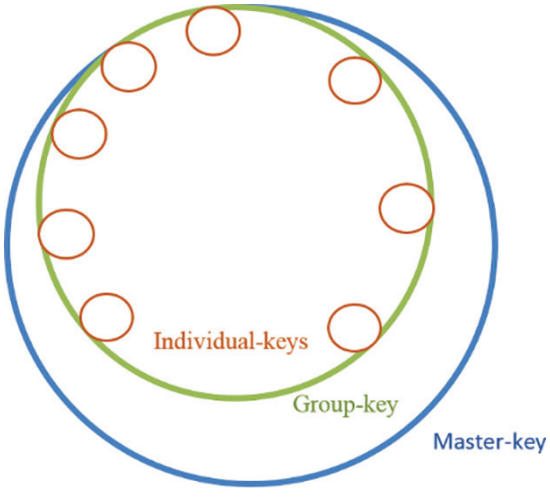

Figure 6: Way of organizing keys in the master-key technique.

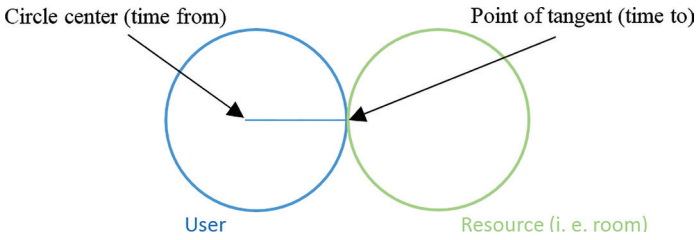

Figure 7: An example of related objects.

the "aging" key, taking advantage of the fact that in both cases, the basis for implementation is the use of certain equations describing geometric figures lying on a common plane. During combining these techniques, we perform two algorithms. The first one will provide each user in the system with their private key. With its help, the user will be able to carry out the available assigned operations. He also gains access to specific resources—depending on the group to which he belongs. An important issue is also respecting the access time. Therefore, it is necessary to run the second algorithm. This will allow the use of an "aging" key mechanism. For example, suppose that the method generating the user's individual key passed the value $y = f(x) = 23x + 42$. Assigning the user a key does not yet entitle him to access any resources. For that, user's circle coordinates must be associated with the target resource. Therefore, let us further assume that the same user has access to the resource from 1/4/2021, at 12:00 PM until 1/8/2021, at 3:30 PM. This means that user is defined by the circle at the point (20210104, 1200). In the next step, the user circle is connected to the circle representing the asset so that they are externally tangent (at point (20210108, 1530))—between these points a straight line is drawn. In this case, it's the line with equation $y_{resource} = f(x) = 82.5x–1667332380$. Let the request be made on 01/05/2021 at 10:55. If the given point belongs to a straight line $y_{resource}$, then the resource-access mechanism is triggered. Note that for this purpose, the user's private key is used, more precisely—the transformation of this key (function isomorphism). Each transaction is associated with a separate isomorphism and parameter x, so that a different key is used in each subsequent request.

V Generalizing the Solution to any Mathematical Function

In our research, the aging-key technique was tested for more advanced functions than a simple linear function. With the principle of the mathematical operations maintained, we obtained the desired result, i.e., an unambiguous and correct answer of the function verifying whether the user has access. However, our approach may be extended to cover any complicated function as necessary, as described below.

In the simplest scenario, one key opens one lock. More complicated situations involve opening the lock with more than one key. For the mechanical solution, it all depends on the number of pins in the lock and the number of indentations that the system blocks or allows its rotation. We assumed that the lock is a function (previously, for the sake of simplifying the problem, it is a linear one) and the lock is opened when it is given the appropriate value of f(x) for a given x. If the number of indentations (unlocking a single pin (Figure 3B)) is greater than one, then each pin indentation (its height) is still a function that can be a transformation of the function f(x) (e.g., a reflection in relation to the OX axis, as already mentioned for the quadratic function).

If we now extend the abovementioned function to any mathematical dependence, we will obtain a very effective tool for the realization of a zero-knowledge proof. Assume that a function is known only to a verified resource. If this resource uses it correctly, i.e., generates a point that meets the specified requirements, then it gains access. The coordinates of the generated point will not allow recreating the form of the function because there are theoretically infinitely many such points. Thus, the appropriately complex function will remain a secret even for a very large number of points generated and presented for the verification. At the same time, while controlling the domain of "correct" and "incorrect" values of a given function (for each of them, there are theoretically endless possibilities), the probability of guessing the "correct" key by coincidence can be easily decreased by manipulating the complexity of the function and its domain.

One may also imagine a situation where the abovementioned function is a set of several or more functions that, when properly parameterized, lead to the same results. This allows for an efficient

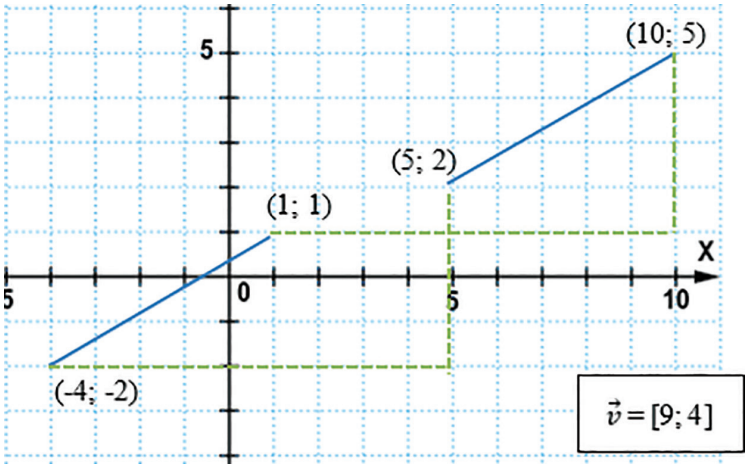

Figure 8: Example of shifting two points of a linear function.

generation of "sub" keys by the owner of the "master" key. For example, each of the subordinate keys can be assigned to a separate straight line segment (i.e., *de facto* the domain of the allowed parameter values and the function result), or the function can be moved by a specific vector (Figure 8). This approach is identical to the group isomorphism [12], except that the target element is not a group but a straight line (which is, in fact, an infinite set of points on a plane).

At this point, we can distinguish several further implementation methods, but in the paper, we will present one of them. Function obtained after performing the shift (by the vector $\vec{v}$) is a onetime key, designated from a private one, which is different for each session/operation. During the communication process, information about the private key is never revealed, but on the basis of the shift and execution of the query for the value (for any x), both parties prove that they have a common secret, without disclosing any information about it. For function with equation $y_1 = f(x) = 4x - 8$ and vector $\vec{v} = [15, 16]$, the session key becomes a straight line with equation $y_2 = g(x) = 4x + 68$. When trying to verify for, e.g., $x = 108$, function y_1 passes value 424, while for function y_2 the result will be equal to 500. With reference to the previous content and the example with the hotel, when accessing the resource (room), the circle representing the user is externally tangent with the circle representing the room, and the straight line between its center and the point of contact can be understood as a straight line y_1 (private key). In response to all subsequent requests, the original function y_1 will undergo transformations. This will act as a generator of the next functions (private keys), allowing the entire verification procedure to carry out differently for each transaction.

When polling the user for, e.g., an access grant, one does not use the assigned key but its transformations. None of the previously obtained functions will reappear for the current session. Their set (transformations of the primary function—for the purposes of the example, it is a linear function, but it can be any mathematical function), can be defined in advance and associated with the user at the very beginning of the transaction. It can also depend on the context, and a common next-shift generator for both sides can be applied.

Earlier examples only referred to the linear function. One of the possible transformations, e.g., the shift operation, will also find application in the case of a quadratic or homographic function. General form of the first one is as follows: $f(x) = ax^2 + bx + c$. For a quadratic function $y_3 = f(x) = 2x^2$ and shift vector vector $\vec{v}_3 = [3; 2]$, the course of the necessary calculations will be as follows:

$$g(x) = \{2[x-(3)]^2\} + 2 \quad (6)$$

$$g(x) = \{2[x^2 - 6x + 9]\} + 2 \quad (7)$$

$$g(x) = \{2x^2 - 12x + 18\} + 2 \quad (8)$$

$$y_4 = g(x) = 2x^2 - 12x + 20. \quad (9)$$

When checking for, e.g., $x = 23$, one obtains a result of the function $y_3 = f(23) = 2 \times 23^2 = 1058$. For the same x, function y_4 will return the value 802. Despite different values for the common parameter x, both values are perfectly correct. Their correctness is always onetime—each iteration uses a different x parameter and a different transformation. Their role can

be associated with a onetime proof of correctness—functions are onetime private keys, while the obtained values provide proof of their possession.

Homographic function is a measurable function of the form $f(x) = \frac{ax+b}{cx+d}$. For a specific function with formula $y_5 = f(x) = \frac{-3x+1}{2x-7}$ and an example vector shift $\vec{v}_4 = [-5; -3]$, of all calculations is limited to:

$$g(x) = \{\frac{-3\left[x-\left(-5\right)\right]+1}{2\left[x-\left(-5\right)\right]-7} + (-3)\} \tag{10}$$

$$g(x) = \{\frac{-3\left[x+5\right]+1}{2\left[x+5\right]-7} - 3\} \tag{11}$$

$$g(x) = \frac{-3x-14}{2x+3} - 3 \tag{12}$$

$$g(x) = \frac{-3x-14}{2x+3} + \frac{-3\left(2x+3\right)}{2x+3} \tag{13}$$

$$g(x) = \frac{-3x-14-3\left(2x+3\right)}{2x+3} \tag{14}$$

$$y_6 = g(x) = \frac{-3x-14-6x-9}{2x+3}. \tag{15}$$

Final form of the function after transformations will be as follows: $y_6 = g(x) = \frac{-9x-23}{2x+3}$. Verification for $x = 42$ will provide the following values for both functions, respectively for $y_5 = -1{,}623376623376623$ and for $y_6 = -4{,}609195402298851$. Despite the use of a completely different function, it applies the same principle of operation. Relying on isomorphisms, we get onetime private keys and one correct value for the parameter x—onetime session/transaction key.

In addition to the transformation involving moving the graph of a function left, right, up, and down and changing its equation, there is a set of other transformations that can be used in this task. These include, among others, symmetrical reflection about the OX axis ($-f(x)$), symmetrical reflection about the OY axis ($f(-x)$), symmetrical reflection about the OX axis of those values that are below it ($|f(x)|$)—transformations related to the module, and combining individual variants.

VI Literature Review

The master-key theme is very popular, confirming the multitude of applications of this technique. In this section, we present a literature review in which the authors of other publications use those techniques mentioned above to solve different but similar research problems.

The authors of the first paper [13] deal with the problem of the exchange of sensitive data among several *IoT* devices. They indicate the need for data protection and rightly note that the involvement of generally available mechanisms and algorithms in a standard form is impossible. They propose a method of key distribution based, among others, on using the streaming-encryption technique and generating random keys based on the master key, followed by a proposal of efficient key distribution to all customers. Our solution depicted in the paper extends the verification by checking whether, in addition to the password, the operation is possible after taking into account, e.g., distance, time of day, or data from sensors from the immediate vicinity. Additionally, we also parameterize session keys.

In [14], the authors focus on big data issues and data gathering from many sources. Due to the possibility of handling vast portions of data in the cloud, one gets some profits related to computation time and obtaining statistics and summaries. Unfortunately, individual data owners may express concerns about information security. The authors propose an algorithm that uses two keys: the main one and the session key provided by the service provider. In this case, the authentication process is based on keys, while the contextual information is bypassed. On the contrary, in our solution, each operation is dynamically associated with a different key (which can be additionally modified by any vector), depending on overall context.

The paper [15] is devoted to ensuring the security of files assigned to successive addresses in the virtual space. The subject certifies the authority to access a given file by presenting a cryptographic pointer referring to the object. A pointer contains a key and a permission structure and is valid if derived from a primary key. Still, the solution limits the scope of the context. In our solution, we additionally solve the problem of an aging (evolving in time/distance) key, and the operation itself does not only result from the data stream sent and accepted by the validator but also from the context.

The next paper [16] also deals with the virtual cloud and security aspects. Due to the trust customers place in the supplier and the ability to effortlessly reach the required information, its protection and security are paramount. Since most services do not support dynamic refreshing of access control and the cost of encryption is high, the issue of the aging key, which we presented in this paper, becomes

important. In our solution, a separate aging key is generated for each operation. During an attempt to perform, e.g., an update to the database, in general, a transaction is allowed after checking whether the user has access (the user and resource circles are externally tangent) and the key is valid (time represented by the radius length is no "longer" than the circle radius). Our approach thus solves the problem of key validity with a single operation.

The topic of the industrial *IoT* (IIoT) has been broadly presented in the article [17]. The data collected by the IIoT has an indisputable dimension and meaning due to its importance of social life. Limited in, among other things, memory, computing power, or energy consumption, end-devices frequently bypass safety obligations. A need to provide a secure communication channel may, depending on the solution, involve long-term master-key exposure and may lead to the disclosure of previous session keys, which is a critical security property. In our solution, each session key can be parameterized and dynamically changed. In addition, our implementation of complex and remote operations may result from the knowledge of the password and the context, e.g., time of the day.

Wireless sensor networks (WSN), widely used for military purposes, intelligent transport, medicine, and authentication and communication issues, are the subject of [18]. Considering the available memory, computing power, or energy demand, the simultaneous application of security measures and an authentication scheme becomes a reason for providing a nonavailable, safe, and efficient way to implement the described problem. The authors of this paper engaged a trusted third party and created an execution environment on the network that acts as a proxy. In the case of two-way data exchange, such a solution will work perfectly, except for huge network traffic. However, this approach fails in the case of devices offering broadcast transmission only (which is typical for many internet of things [IoT] devices). On the contrary, our approach deals with both unicast and broadcast traffic.

The method of data protection in which, based on the master key, it is possible to generate a derivative key and perform encryption operations is described in the paper [19]. The proposal is directed as an inner product functional encryption (IPFE) scheme in which the problems of an attack with a selected chosen-ciphertext attack (CCA) and an attack with a selected plaintext chosen-plaintext attack (CPA) are solved. In our solution, such attack types are not possible, as a single-usage key is generated for each operation, and each time.

These abovementioned papers concern security issues that require extraordinary measures with a constantly growing interest in using the growing possibilities of the *IoT*. Although these techniques are not always possible to implement (e.g., due to limited memory or energy consumption), each system designer is still responsible for selecting and applying an appropriate data protection method. Thus, our solution fills the gap, providing an easy-to-implement approach capable of being applied for limited hardware.

VII Conclusions

The paper presents a new way of organizing user access to resources using the master-key technique and the aging-key technique. In the first case, with a single key, a user, depending on the rights, gains the ability to perform many actions within a certain context. Using geometric calculations (transformations), and more specifically, the properties of the circles and the ways of their mutual position, in our solution, we identify available points inside the circles (symbolizing, e.g., a room, a device, an action to be performed, etc.). If the circle representing the user has a tangent point with another circle, which represents the resource, such as these circles are externally tangent, then the user has granted access to this resource. A straight line is defined between the center of the user's circle and the tangent point with another circle, related to the access time. Thanks to it, the aging-key technique was successfully introduced, for which the verification depends on an independent variable (e.g., a passage of time). The paper also presents the purpose of this technique in relation to the possibility of using any function *f(x)* and any transformations of this function. Sets of functions and their transformations may depend on context, location, and other parameters, and each modification is, in fact, an operation performed on the original circle equation and the line connecting it to the resource.

Going to the examples of applications, it is worth noting that the prepared solution was initially intended for *IoT* devices, for which the primary form of interaction is *ad hoc* data exchange, in random places and situations. The use of the extended verification described in the paper solves the problem of strong contextual verification, which requires the processing of many parameters to determine access rights. However, the possible applications are not limited to such a narrow area. First of all, the presented solution can be successfully implemented wherever there is a hierarchical employee structure and individual/group

access to resources (houses, blocs, companies, office buildings, public utility facilities, shopping centers, hotels, dormitories, schools, etc.). For such a complex environment, a classical certificate is an insufficient set of data to be able to determine with certainty whether a given action is possible in the current context, such as only at duty hours, with a physical presence of a person at certain place, etc. For such complex cases, the certificates without additional metadata defining the context do not entirely fulfill their role. Performance of an action/operation may depend on the role (lecturer/student), the time elapsed, a period of a day (e.g., is it dark enough to turn on the light), as well as distance or any other parameters of the external context. A substantial advantage of the proposed solution is the fact that such verification does not require additional procedures, usually carried out in other ways than cryptography.

References

[1] Gupta, Purnima & Sinha, Amit & Srivastava, Prabhat & Perti, Ashwin & Singh, Aswani. (2021). Security Implementations in IoT Using Digital Signature. 10.1007/978-981-15-4692-1_40.

[2] Ferguson, Niels & Schneier, Bruce & Kohno, Tadayoshi. (2015). Message Authentication Codes. 10.1002/9781118722367.ch6.

[3] Galbally, Javier & Marcel, Sébastien & Fierrez, Julian. (2014). Biometric Antispoofing Methods: A Survey in Face Recognition. IEEE Access. 2. 1530-1552. 10.1109/ACCESS.2014.2381273.

[4] (2021). One Time Password (OTP) Life Cycle and Challenges: Case Study. 10.1007/978-981-15-7961-5_13.

[5] Rubinstein-Salzedo, Simon. (2018). Zero-Knowledge Proofs. 10.1007/978-3-319-94818-8_16.

[6] Wollinger, Thomas & Kumar, Sandeep. (2006). Fundamentals of Symmetric Cryptography. 10.1007/3-540-28428-1_9.

[7] Ranjan, Rajiv & Mukherjee, Abir & Rai, Pankaj & Ahmad, Khaleel. (2019). Asymmetric Cryptography. 10.1201/9781351021708-8.

[8] Ferguson, Niels & Schneier, Bruce & Kohno, Tadayoshi. (2015). Block Ciphers. 10.1002/9781118722367.ch3.

[9] Bray, Shannon. (2020). Stream Ciphers and Block Ciphers. 10.1002/9781119612216.ch5.

[10] Hawanna, Varsharani & Kulkarni, Vrushali & Rane, Rashmi & Mestri, P. & Panchal, S.. (2016). Risk Rating System of X.509 Certificates. Procedia Computer Science. 89. 152-161. 10.1016/j.procs.2016.06.027.

[11] Rhodes, Brandon & Goerzen, John. (2014). TLS/SSL. 10.1007/978-1-4302-5855-1_6.

[12] Ferrer, Marita & Gary Gutiérrez, Margarita & Hernández, Salvador. (2019). Representation of group isomorphisms I. Topology and its Applications. 263. 10.1016/j.topol.2019.05.023.

[13] Jassim S.A., Farhan A.K., Combined Chebyshev and logistic maps to generate pseudorandom number generator for internet of things (2022), International Journal of Electrical and Computer Engineering, 12 (3), pp. 3287-3297, DOI: 10.11591/ijece.v12i3.pp3287-3297

[14] Shrihari M.R., Manjunath T.N., Archana R.A., Hegadi R.S., Development of Security Performance and Comparative Analyses Process for Big Data in Cloud (2022), 789, pp. 147-160.

[15] Lopriore L., Cryptographic pointers for fine-grained file access security (2022), DOI: 10.1080/19393555.2022.2033365

[16] Susilabai S.S., Mahendran D.S., Peter S.J., A Trusted User Integrity-Based Privilege Access Control (UIPAC) for Secured Clouds (2022), 243, pp. 499-520, DOI: 10.1007/978-981-16-3675-2_38

[17] Fan Q., Chen J., Shojafar M., Kumari S., He D., SAKE*: A Symmetric Authenticated Key Exchange Protocol with Perfect Forward Secrecy for Industrial Internet of Things (2022), DOI: 10.1109/TII.2022.3145584

[18] Liu X., Guo Z., Ma J., Song Y., A Secure Authentication Scheme for Wireless Sensor Networks Based on DAC and Intel SGX (2022), 9 (5), pp. 3533-3547, DOI: 10.1109/JIOT.2021.3097996

[19] Liu X., Liu S., Han S., Gu D., Tightly CCAsecure inner product functional encryption scheme, (2022) Theoretical Computer Science, 898, pp. 1-19.