

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/377074406>

# One-Time Passwords: A Literary Review of Different Protocols and Their Applications

Conference Paper in Communications in Computer and Information Science · January 2024

DOI: 10.1007/978-3-031-48855-9\_16

CITATIONS

2

READS

1,193

6 authors, including:



Luis Almeida  
National Polytechnic School

4 PUBLICATIONS 10 CITATIONS

SEE PROFILE



Brayan Fernandez  
National Polytechnic School

2 PUBLICATIONS 3 CITATIONS

SEE PROFILE



Anthony Almachi  
National Polytechnic School

2 PUBLICATIONS 3 CITATIONS

SEE PROFILE



Sang Guun Yoo  
National Polytechnic School

96 PUBLICATIONS 1,478 CITATIONS

SEE PROFILE

# One-Time Passwords: A Literary Review of Different Protocols and Their Applications

Luis E. Almeida<sup>1,2</sup>, Brayan A. Fernández<sup>1,2</sup>, Daliana Zambrano<sup>1,2</sup>, Anthony I. Almachi<sup>1,2</sup>, Hilton B. Pillajo<sup>1,2</sup> and Sang Guun Yoo<sup>1,2,3,\*</sup>[0000-0003-1376-3843]

<sup>1</sup> Departamento de Informática y Ciencias de la Computación, Escuela Politécnica Nacional, Quito, Ecuador

<sup>2</sup> Smart Lab, Escuela Politécnica Nacional, Quito, Ecuador

<sup>3</sup> Departamento de Ciencias de la Computación, Universidad de las Fuerzas Armadas ESPE, Sangolquí, Ecuador

\* Corresponding author

sang.yoo@epn.edu.ec

**Abstract.** Currently, user authentication only based on passwords can be inadequate due to different types of vulnerabilities and attacks. To solve this problem, two-factor authentication is commonly used, providing a higher level of security when the user logs into their accounts, and one popular example of two-factor authentication is the combination of password and One-Time Passwords (OTP). Due to the importance and popularity of OTPs, this study analyzed the most widely used OTP protocols and their applications to understand their state of the art. The scientific community can use the analysis carried out in this work to understand why OTP is so popular and to decide on the type of OTP, in case a custom implementation is needed for an authentication system. To achieve this, this work analyzed a large number of previous works methodically through a semi cyclic process based on research action combined with a systematic review process. The most important works were analyzed to identify their specific features and to classify the used technologies. Usage trends in terms of protocols, implementations, algorithms, and OTP generators were also analyzed. In addition, this article has determined a complementary feature guide that must be considered when implementing an OTP authentication system.

**Keywords:** Authentication process, One-Time Passwords, OTP, OTP protocols, OTP generators, Two-Factor Authentication, 2FA.

## 1 Introduction

Security in different technological areas, such as the Internet of Things [1], networks [2] and software development [3], has given great importance to the authentication process. This process aims to establish trust between users and devices, verifying the identity of users on a platform [4]. Although the use of username and password has been common for authentication [5], the advancement of the Internet has made identity theft a significant security problem [6].

Based on research from the last 30 years, a strong single factor authentication based on passwords has been found to be difficult to implement due to various threats and attacks [7], such as compromised devices with spyware, intercepted communications, brute force attacks, and "Man in the Middle" attacks [8][9]. As a result, two-factor authentication (2FA) has gained popularity as a more effective security measure [10]. 2FA combines two of the three universal factors: something the user knows, something the user has, and something the user is or does [11]. The key advantage of 2FA is that even if one factor is compromised, the security of the system is maintained [11].

Among the various ways to implement 2FA, the use of One-Time Passwords (OTP) based on software tokens is one of the most popular methods [12][13]. These programs use OTP algorithms for their implementation and are fundamental to guarantee the security of systems in applications of online banking, electronic commerce, medical care, IoT and other scopes [12][13]. Lamport [14] proposed the first OTP known as the S/KEY authentication system [15], to authenticate untrusted computers in public use.

Due to the advances in the use of OTP-based systems, this study focuses on analyzing different types of OTP and their algorithms, as well as their respective characteristics, in order to generate a complete document on the state of the art of this technology.

## **2 Research Methodology**

The primary objective of this research is to analyze the current trend regarding OTPs implemented in different solutions. To achieve this, a semi cyclic process based on research action [16] and combined with a systematic review process [17], [18] has been applied. This methodology is the same used in [19]. Each of these phases has specific tasks executed during the research development process (see Fig. 1).

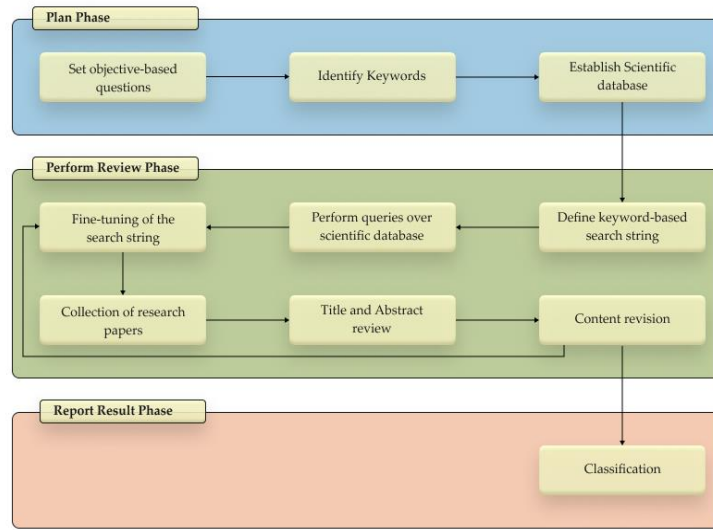
### **2.1 Planning Phase**

This is a preparatory phase prior to starting the research process. The main objective is to reduce the scope and produce accurate searches that are suitable for the present study. To achieve this, the following research questions were proposed: (1) what are the solutions in the current implementations of OTP?, (2) what kinds of algorithms are used in the different implementations of OTP?, and (3) in what areas are OTPs used?

With the defined questions, the following keywords were identified: "One-Time Passwords", "solutions", "implementations", "2FA", "areas". Subsequently, the preliminary search strings were "One-Time Password solutions", and "One-Time Password implementations". These strings resulted in large amounts of documents, and to reduce the number of documents, we used logical connectors to combine the previously mentioned keywords. Additionally, scientific databases were selected to search for articles using the previously defined strings. The selected databases were ACM Digital Library, IEEE Xplore, Springer, and ScienceDirect.

## 2.2 Perform Review Phase

In this stage of the study, searches were carried out in the digital repositories using specific previously defined strings. Only publications older than 5 years are considered, with some exceptions, as mentioned in the definitions of [18] and [19], because certain algorithms used in the development of solutions based on OTPs were developed in longer than the specified period of time.



**Fig. 1.** Detailed research method.

For the collection and management of the articles, the Mendeley tool was used, which allows sharing and managing research papers through labels and filters. In this way, a shared library could be created to facilitate the analysis of the bibliography.

To conclude this stage, an article discard protocol was applied manually. The titles and abstracts of each article are reviewed to identify relevant keywords and details that might answer the research questions posed above. Those documents that did not comply with the terms or did not provide relevant information were discarded.

## 2.3. Report Result Phase

In the last step of the research methodology, the findings and results were documented and used to build the following sections, which are the central part of this work. Furthermore, new OTP technological solutions and their implementation areas were analyzed and discussed. The results of this phase are shown in the following sections of this document.

### 3 One-Time Passwords

OWASP Top 10 indicates that an authentication system may be exposed to different types of threats [20]. This situation shows how important is the implementation of 2FA based on OTP. When a system implements only the password-based authentication systems, it can be exposed to different types of attacks. For example, we could mention the attack executed on the PlayStation Network, where a group of attackers gained access to 77 million customer accounts, including credit card information [21].

The study of OTPs began with Lamport [14] in the early 1980s, with a protocol in which both the client and the server agreed to use an algorithm to generate OTPs, which expired once the authentication process was successfully carried out. Lamport's solution used a seed ( $S$ ) agreed upon by both the client and the server, which passed through a hash function ( $h$ )  $n$  number of times. However, this presented a problem, since after a certain number of repetitions, hash functions tend to repeat the output values [22].

Subsequently, other OTP solutions emerged, such as the work presented in [23]. In this work, the protocol is based on a counter ( $C$ ) that increments its value and is applied to a hash chain with a key for message authentication (HMAC). The counter on the server increments each time a successful authentication is achieved, while for the user it increments each time a new OTP value is required. In another work, an improvement of [23] is proposed by implementing the time factor instead of the counter, which avoids the desynchronization between the server's and client's counter [24]. In [24], timestamp marks are used to generate the values to be sent to the server to achieve the authentication process. The synchronization, both for the client and the server, makes use of the Unix time, as it is universally used by Unix type operating systems.

#### 3.1 Protocols for generating One-Time Passwords

Today, OTPs are a common authentication mechanism for many companies, institutions, and even governments looking to upgrade their security strategy. For example, Google uses the sending of OTP via SMS to authenticate the user after numerous failed login attempts. On the other hand, telecom companies generate a One-Time Password and send it directly to the user's mobile phone as an authentication privilege for using the free internet service in public places such as shopping centers, maritime terminals, or airports, which has a validity of 30 minutes [25].

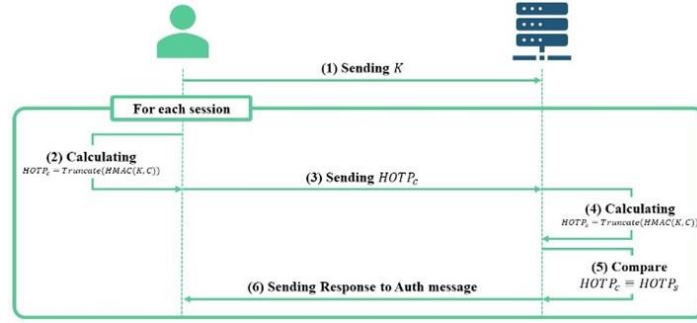
OTP algorithms can be classified into two groups, i.e., event based and seed value-based OTPs. Event based OTPs change value whenever an event occurs. The main event-based OTPs are HOTP and TOTP. In the case of HOTP, if the counter value changes by a login attempt event, a new HOTP value is generated. Similar process is done in TOTP, where a new value is generated when the time value changes, invalidating the previous value. On the other hand, seed value-based OTPs are based on a shared seed that is used to generate the OTP value.

##### 3.1.1. HMAC-Based One-Time Password Protocol (HOTP)

HOTP is an event-based OTP algorithm created in 2005 that generates values based on hash message authentication codes (HMAC) [26], [27], [28]. For generating HMACs, the Secure Hash Algorithm-2 (SHA-2 or SHA2) is used frequently among the different hash algorithms. However, in 2015, NIST recommended the use of Secure Hash Algorithm-3 (SHA-3 or SHA3) due to known weaknesses that SHA-2 has exhibited over time [27], [28].

HMAC is used to create the HOTP value. The HOTP algorithm works based on an increasing counter value (C) and a static symmetric key (K) known only to the token and validation service [29]. The key (K) must be shared between the client and the server, and the counter (C) must be synchronized between the HOTP generator (client) and the HOTP validator (server) [30][31].

HOTP works with two counters, one on the client side (hardware or software token) and one on the server side. Both counters validate the HOTP value. The server increments its counter after each successful authentication, while the client does so when requesting a new OTP. If the server receives a value that does not match that of the client, the resynchronization protocol is started before another attempt [23], this process can be seen summarized in Fig. 2.



**Fig. 2.** HOTP algorithm protocol, adapted from [22]

For the user to easily remember the OTP, the most common OTP length is 6 digits, resulting in ten million possible combinations. HOTPs are vulnerable to brute force attacks because their value expires only after successful authentication [23]. In [23], two solutions are proposed to detect and stop brute force attacks on the authentication server: (1) Define a maximum number of possible attempts for HOTP validation and (2) Implement a delay scheme to avoid Multiple parallel divination techniques. For each failed attempt at a login session, the authentication server would wait for: (Number of failed attempt) \* (Definite time).

HOTP has no expiration time and can be used for a long period of time, increasing the chances of being attacked. This weakness is solved by the TOTP algorithm, which provides short-lived OTP values to improve security [24].

### 3.1.2. Time-Based One-Time Password (TOTP).

Time-based One-Time password (TOTP) is an extension of the HOTP algorithm. The main difference is that the HOTP uses a counter (C) while the TOTP uses time (T). The time T can be defined as:  $TOTP = HOTP(K, T)$ , where T represents the number of time steps between the initial counter time  $T_0$  and the current Unix time (i.e., the number of seconds elapsed since midnight UTC of January 1, 1970) [24]. In general, the time value tends to be the date (YYYY-MM-DD) followed by the time (HH:MM:SS). Both are usually measured in Coordinated Universal Time (UTC) or as well-known as Greenwich Mean Time (GMT) because it can be used depending on the location or the time zone [27], [28].

The HOTP method requires a timer. The validation system receives the TOTP value without knowing the exact OTP generation timestamp, which creates a gap between generation and reception. To address this, an authentication policy is established with an acceptable transmission delay window. A time step of 30 seconds is recommended as a proper balance between security and usability [24].

TOTP is based on the HOTP algorithm, which is derived from HMAC and uses defined hash functions. That is, part of the security of the TOTP depends on the hash functions used in the HMAC algorithm. Consequently, TOTP implementations choose to use HMAC-SHA-256 or HMAC-SHA-512 instead of the HMAC-SHA-1 functions [24].

### 3.1.3. Seed Value-Based One-Time Password

In [32], an example of seed-value-based One-Time Password is described. To generate the OTP value, three entities are utilized: user, server, and One-Time Password Generation System (GS). The GS receives data from both the user and the server to generate the OTP. The user provides a number (N) and a secret password phrase (PP), while the server provides a seed (S). The GS uses the user's secret password phrase along with the seed received from the server, applying the secure hash function N times to generate a sequence of OTP values, equation 8 describes the formula for generating the OTP.

$$OTP = Hash_{function}(PP, S)^N$$

The seed comes from the server in a clear text which is purely alphanumeric of 1 to 16 characters long that must be internally converted to lower case. As with the other OTP algorithms described before, the security of seed value-based OTP also depends on the used hash function.

Another way to use a seed value-based OTP is using pseudorandom numbers. A pseudorandom number generator (PRNG) is commonly used to generate unpredictable OTP values [32]. The pseudo random numbers are values or elements statistically random, derived from a seed [32]. A cryptographically strong pseudorandom number generator is needed by the server to generate the OTP value for each login session [33].

The security of the PRNG depends on the random algorithm used, but also requires a correct implementation, for example, choosing a constant as the seed for the random algorithm, the PRNG could be predictable. For that reason, there are critical randomness rules [3]: (1) do not use a constant or predictable seed to initialize the random function, (2) do not use a static OTP value, and (3) do not generate OTP values according to specific patterns.

In the previous description, we get a wider idea of how OTPs are implemented. Depending on the required complexity and the requirements set by the organization, we can choose one or the other. Nevertheless, we can see that the trend of these protocols is divided into: Time-based OTPs, Hash-chain based OTPs, and Challenge-based Authentication algorithms. Table 1 shows a summary of the type of OTP used in the reviewed articles. The structure of this table is as follows, the first column presents the different types of OTP protocols analyzed in each article, while the second column lists the corresponding articles. In the last row of the table are articles that do not specify the type of OTP protocol used. However, these articles provide valuable information about the operation and applications of OTPs, which allows for a more complete understanding of the topic.

**Table 1.** OTP Classification.

Type of OTP	References
Time-based OTP	[7] [19] [25] [29] [30] [35] [36] [38] [39] [40] [47] [53] [55]
HMAC based OTP	[7] [12] [14] [19] [25] [29] [34] [41] [42] [43] [44] [45]
Seed value base OTP	[2] [3] [11]
Not specified	[1] [5] [37] [46] [48] [49] [50] [51] [52] [54]

#### 3.1.4. Other OTP Generation Protocols

Although they are not very common, some previous works make use of other types of protocols. The following is a list of some of them:

**YSH Protocol:** The YSH protocol is an enhancement of the Lamport hash chain protocol for generating One-Time Passwords (OTP). Although it allows the server to be verified, it does not protect against spoofing attacks by not storing certain client parameter values, which compromises authentication. Furthermore, it is vulnerable to "small number attack" [15].

**Bicakci Protocol:** The protocol is based on asymmetric cryptography to generate and verify OTPs, being an evolution of the HOTP protocol to improve security. However, it has disadvantages such as computational complexity and lack of server authentication, which could allow for phishing attacks [22].

**Chefranov's Scheme:** The protocol features a complex and secure algorithm with many parameters. However, its high complexity makes it difficult to implement and understand in comparison with previous protocols. Although it is resistant to browsing attacks, it lacks protection against check table modification. Furthermore, the Chefranov scheme allows for server verification, but does not protect against spoofing [22].

### 3.2 Methods of Receiving One-Time Passwords

The previous works analyzed in this paper shown different ways of sending and receiving the OTPs. In the following, there are the explanation of those methods.

**Text Messages:** The common method to deliver OTP is via SMS [3]. However, studies such as [32], [34], [35] have shown that SMS is vulnerable to attacks such as SIM card cloning and shoulder surfing. Despite this, companies like Airbnb, Facebook, and Google continue to send OTPs over SMS due to user preference [5]. Alternatives using lightweight cryptography and text steganography have been also proposed [36].

**Quick Response Code (QR):** According to [37], scanning QR codes is a relevant method to receive OTP which allows saving time by avoiding manual input of characters. Although the average authentication time is 25.8877 seconds, [38] highlights that QR codes are vulnerable to replacement, modification, and malicious URL attacks.

**Piezo-Gyro Channel:** A work proposed by [39] presents a unique way to input an OTP using a physical device that automatically sends the OTP to a smartphone through acoustic stimuli on its MEMS gyroscope. The device uses a piezoelectric transducer to induce movements in the internal mass of the gyro sensor, generating artificial angular velocity readings that are translated into the OTP value. This process creates a one-way communication channel between the physical device and the smartphone, allowing authentication without the user having to manually enter the OTP [39].

**Proprietary Tokens:** Proprietary tokens are small physical devices, such as hardware keys. They do not require passwords but may require a physical or wireless connection to authenticate the user. In addition to cards, there are other tokens such as USB and wireless devices. Cards are more likely to be lost and cloned [40].

**Paper:** The OTP paper receipt method is a form of authentication that uses a list of printed codes to securely transmit information. It is useful in situations where access to devices is complicated or impossible [42]. For example, Google verification codes work like this. Ten one-time passcodes are generated when 2 Step Verification is enabled on a Google account, and new codes are generated when old ones are used up [43].

**Email:** Email is a popular way to send OTPs due to its ease of use, speed, and cost effectiveness. According to [25], they propose a virtual OTP keyboard represented as a 4x4 matrix with XY coordinates sent to the user to select on the virtual keyboard. Email based OTPs are secure and allow user identity to be authenticated during login on any Internet enabled device. However, [44] warns that they can be intercepted or compromise the user's account.

### 3.3 Classification of the use of One-Time Passwords

One-time password (OTP) passwords are a security tool used to protect information and ensure user authentication in different applications and systems [45]. These passwords are generated randomly and can only be used once, making them an effective security measure against unauthorized access [8]. OTP passwords are commonly used in the access of online bank accounts [46], user authentication in email services [40], and access to private networks and two-factor authentication systems [2], [47]. However, as of the writing of this document, some works were found that utilize OTPs in different ways and in different areas as detailed below.

**Use of OTPs in Mobile Applications:** Currently, mobile applications have had great success due to their portability and efficiency [5]. One area that has taken advantage of this is online banking, with the implementation of virtual wallets [5]. A study

focused on Human-Computer Interaction (HCI) and Computer Supported Cooperative Work (CSCW) of the digital payment apps Paytm and PhonePe was carried out, too [48]; both apps use OTPs to perform the login or user registration process by sending a unique code to the user's registered mobile number [48]. To make a transaction in the Paytm app, a unique QR code must be scanned for each store. However, in several countries, card payments, including those made through digital wallet apps, require a two-factor authentication (2FA) [48]. Both the CVV (the three-digit code on the back of the card) and an OTP delivered to the user via the bank's own app or SMS must be entered [48]. Although methods vary among banks, for the case studied in [48], it indicates that the user has 180 seconds or less to complete the authentication process.

**Use of OTP in Hospitals and Healthcare Centers:** In [5], TreC, a platform developed in Northeast Italy to manage personal health records (PHR) is described. In addition to the main web system, TreC offers specific solutions for chronic patients, such as TreC Diario Diabete, which allows recording health data such as blood glucose level and physical activity, and TreC FSE, which allows access to personal health data and medical prescriptions from the phone.

Authentication in TreC is based on a Multi-Factor Authentication (MFA) system that uses One-Time Passwords (OTP). These codes are displayed on the screen of the patient's mobile device and entered into the web system to validate her identity. This OTP authentication approach has also been used in hospital settings, such as emergency care and vehicle services, where tablets enabled with insecure passwords are employed in emergency vehicles to quickly access necessary information.

The use of OTP is essential to protect the patient information payload in the hospital system and prevent cyber attacks that may reveal private information. Many hospital systems are outdated and use outdated operating systems, making them vulnerable to malware attacks. In addition, the lack of personal computers for medical personnel leads to the sharing of potentially infected computers with keyloggers.

The COVID-19 pandemic presented additional challenges as hospitals had to perform tests and deliver results with precision and privacy. It was proposed to use tokens to assign results in electronic government systems, but this could increase costs and exclude people with limited resources or without access to the Internet.

Finally, it is highlighted that people with mental disabilities also use OTP to generate secure access codes to systems, which allows them to authenticate or validate their access through specific applications or devices [50].

**Usage of OTP in IoT:** In an Internet of Things (IoT) system [51], an approach to improve the security of smart locks using one-time passwords (OTP) is proposed. In this system, when someone rings the doorbell of an apartment equipped with a smart lock, the owner, Bob, receives a notification on his phone. In addition, a camera built into the doorbell takes a photo of the person who has rung the doorbell and sends it to Bob.

To verify the identity of the person, Bob uses the photo to confirm that it is Alice. Instead of sending him the password that he always uses, Bob generates a temporary password (OTP) and sends it to Alice. In this way, Bob keeps the original password for his apartment private, improving the security of the system.

## 4 Discussion

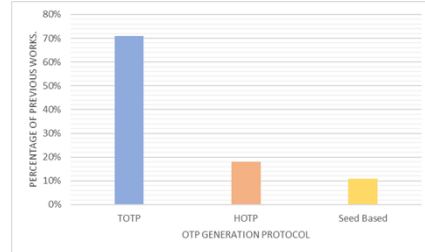
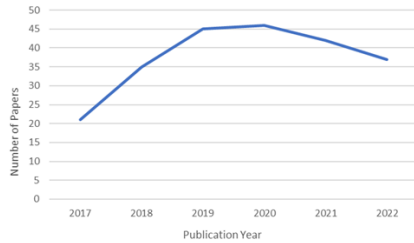
As shown in Fig. 3, research related to OTPs experienced constant growth from 2017 to 2020, with 2019 and 2020 being the years of greatest research activity. However, starting in 2020, there was a decrease in the number of published works, when we were overcoming the COVID-19 pandemic. It is possible that the priority attention towards projects related to the pandemic has affected the interest in the OTPs. The need for future research is raised to better understand the trend in this area, especially in authentication issues which is of great relevance in computer security.

### 4.1 One-Time Password Generation Protocol

71% of the relevant research projects focus on the TOTP (Time-Based One-Time Password) protocol for implementation or highlight its use. According to references [33] and [3], TOTP is considered superior to the HOTP protocol in terms of robustness for generating one-time passwords (OTPs). The factors that contribute to this greater robustness are the following:

- **OTP Randomness:** TOTP uses time as the key to generate cryptographically strong pseudo-random values using HMAC, which results in completely random one-time passwords. This increases the randomness of the one-time passwords generated by TOTP, as mentioned in article [33].
- **OTP Expiration:** By default, article [39] specifies that after  $x$ , a determined time, a new OTP value will be generated (e.g., a recommended time of 30 seconds). This prevents brute force attacks. However, it is possible that values generated once the time span has expired may still be valid for a short period due to latency and other communication-affecting effects, as mentioned in [3].
- **OTP Consumption:** Although not a TOTP-specific feature, several systems using two-factor authentication do not limit the number of allowed incorrect OTP entry attempts, as mentioned in article [33]. This can lead to excessive usage of OTPs.

According to Fig. 4 in the research papers, a trend is observed in the OTP generation protocols. The TOTP protocol has the highest percentage of implementations at 71%, followed by HMAC-based One-Time Password (HOTP) at 18%, while value-based OTP implementations are at 11%.



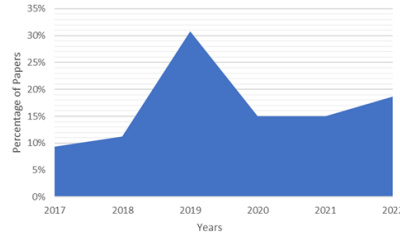
**Fig. 3.** Trend of number of previous works related to OTP.

**Fig. 4.** Usage of OTP Protocols.

However, there are concerns about the security of one-time password generation due to possible poor implementation practices. For example, the use of basic random number generation functions or predictable initial values could compromise security. An attacker could obtain a bank of values and predict the next value in the sequence if he knows the seed, allowing the entire sequence of values to be generated.

The article [33] mentions that many programming languages have predefined random number generation functions, but these are usually not cryptographically secure. Therefore, it is essential to keep this in mind to ensure the security of one-time password generation.

Fig 5. shows that the TOTP protocol peaked in publication in 2019, correlating with the data in Figure 13. However, in 2020, there was a 16% decrease in TOTP-related publications. This could be because some articles mention the use of one-time passwords without making a clean reference to the TOTP protocol.



**Fig. 5.** OTP uses over the years.

## 5 One-Time Password Selection Strategy

In the context of current research, several features of OTP algorithms have been identified. It is noted that all solutions use hash functions to add an additional layer of security, ensuring that an attacker cannot obtain the original value even if they manage to obtain the resulting hash value.

It is mentioned that HOTP may be vulnerable to brute force attacks once the OTP value is generated and expires on successful validation, potentially allowing an attacker to determine the value if there is no limit on the number of authentication attempts. failed.

In the case of TOTP, it is mentioned that the vulnerability is reduced thanks to the use of the time value for each OTP value, which disables it after a short period of time and makes a brute force attack almost impossible.

In addition, it is noted that seed-based one-time passwords can be vulnerable to network snooping attacks, where an attacker obtains seed values to generate OTPs and uses them to steal the shared secret or launch attacks.

Despite the aforementioned drawbacks, each OTP generation protocol has ideal scenarios for its implementation. HOTP and seed-based OTP are suitable for systems

where the OTP value does not need to be updated quickly and where the level of security is not critical, or where ease of use is a priority. On the other hand, the TOTP is recommended for systems where security is a primary concern, as it limits the time to enter a 6-digit value to just 30 seconds.

When selecting a one-time password (OTP) it is important to adhere to its ability to meet appropriate security standards [33]. This involves choosing an OTP with a strong cryptographic algorithm and additional security measures. Ease of use and compatibility with existing systems and devices [36] should also be considered, as a difficult-to-use or incompatible OTP can result in low adoption and lower security.

Furthermore, the durability of the OTP value is relevant. An OTP that is resistant to reverse engineering and not easily compromised should be chosen [33], since an OTP that is susceptible to attacks or short-lived might not be suitable for long-term use.

As for the methods of receiving OTP, each has advantages and disadvantages. Paper OTP cards are easy to use and do not require electronic devices or internet connection. However, there is a risk of losing, duplicating, or discarding these cards, which could allow an attacker to authenticate illegitimately.

QR codes are used in messaging applications to authenticate devices when logging in, improving the experience through the use and communication of One-Time Passwords (OTP) without the need to enter codes manually, which increases security. However, some devices without cameras limit accessibility, and security could be compromised if attackers were to steal images with QR codes that contain OTP. Additionally, it is important to consider the cost and availability of OTPs before choosing them, as they may not be feasible for certain organizations. [40].

## 6 Conclusions

According to the reviewed literature, there are various types of protocols that allow us to make use of OTPs. A clear trend in the use and implementation of the protocols established in different RFCs, such as HOTP, TOTP and Seed value-based OTP was evidenced in the reviewed works.

To achieve a satisfactory authentication process, different OTP protocols can be used. However, the selection and usage conditions of these protocols depend on the advantages and communication frequency between the OTP generator server and the user, as assessed in previous sections.

To effectively implement different types of one-time passwords (OTP), it is important to follow certain recommendations. First, it is essential to ensure that the chosen OTP complies with the appropriate security standards and is compatible with existing systems and devices. It is important to perform compatibility and security tests before implementing the OTP to ensure that it meets the organization's needs.

While the protocols previously discussed the focus on security and increased robustness of different One-Time Password implementations, the user experience of entering the passwords where required is often neglected.

In conclusion, to effectively implement different types of OTPs, it is important to ensure that they comply with appropriate security standards, are compatible with

systems and devices, and that there is a balance between the cost and benefit of implementing a two-factor authentication system.

In terms of future work, it is proposed to conduct a study that helps to determine which type of protocol and delivery method the user feels most comfortable with, as well as to implement a solution that makes use of the most widely used protocols.

## References

1. W.-C. Tsai, T.-H. Tsai, T.-J. Wang, and M.-L. Chiang, "Automatic Key Update Mechanism for Lightweight M2M Communication and Enhancement of IoT Security: A Case Study of CoAP Using Libcoap Library," *Sensors*, vol. 22, no. 1, p. 340, Jan. 2022.
2. X. Zhou, Y. Lu, Y. Wang, and X. Yan, "Overview on Moving Target Network Defense," in 2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC), Jun. 2018, pp. 821–827.
3. S. Ma et al., "Fine with '1234'? An Analysis of SMS One-Time Password Randomness in Android Apps," in 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), May 2021, pp. 1671–1682.
4. J. Zhang, X. Tan, X. Wang, A. Yan, and Z. Qin, "T2FA: Transparent Two-Factor Authentication," *IEEE Access*, vol. 6, pp. 32677–32686, 2018.
5. G. Sciarretta, R. Carbone, S. Ranise, and L. Viganò, "Formal Analysis of Mobile Multi-Factor Authentication with Single Sign-On Login," *ACM Transactions on Privacy and Security*, vol. 23, no. 3, pp. 1–37, Aug. 2020.
6. S. Ruoti and K. Seamons, "End-to-End Passwords," in *Proceedings of the 2017 New Security Paradigms Workshop*, Oct. 2017, pp. 107–121.
7. D. Wang, W. Li, and P. Wang, "Measuring Two-Factor Authentication Schemes for Real-Time Data Access in Industrial Wireless Sensor Networks," *IEEE Trans Industr Inform*, vol. 14, no. 9, pp. 4081–4092, Sep. 2018.
8. F. Aloul, S. Zahidi, and W. El-Hajj, "Two factor authentication using mobile phones," in 2009 IEEE/ACS International Conference on Computer Systems and Applications, 2009, pp. 641–644.
9. M. Shirvanian and S. Agrawal, "2D-2FA: A New Dimension in Two-Factor Authentication," in *Annual Computer Security Applications Conference*, Dec. 2021, pp. 482–496.
10. K. Aravindhan, "One-time Password: A Survey," *International Journal of Emerging Trends in Engineering and Development Issue 3*, vol. 1, no. 3, 2013.
11. N. Gunson, D. Marshall, H. Morton, and M. Jack, "User perceptions of security and usability of single-factor and two-factor authentication in automated telephone banking," *Comput Secur*, vol. 30, no. 4, pp. 208–220, Jun. 2011.
12. E. Erdem and M. T. Sandikkaya, "OTPaas—One Time Password as a Service," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 3, pp. 743–756, Mar. 2019.
13. C. Jin, Z. Yang, M. van Dijk, and J. Zhou, "Proof of aliveness," in *Proceedings of the 35th Annual Computer Security Applications Conference*, Dec. 2019, pp. 1–16.
14. L. Lamport, "Password authentication with insecure communication," *Commun ACM*, vol. 24, no. 11, pp. 770–772, Nov. 1981.
15. W.-B. Leea, T.-H. Chen, W.-R. Sun, and K. I.-J. Ho, "An S/Key-like One-Time Password Authentication Scheme Using Smart Cards for Smart Meter," in 2014 28th International Conference on Advanced Information Networking and Applications Workshops, May 2014, pp. 281–286.

16. J. S. Drummond and M. Themessl-Huber, "The cyclical process of action research," *Action Research*, vol. 5, no. 4, pp. 430–448, Dec. 2007.
17. S. Chauhan, N. Agarwal, and A. K. Kar, "Addressing big data challenges in smart cities: a systematic literature review," *info*, vol. 18, no. 4, pp. 73–90, Jun. 2016.
18. C. M. de Morais, D. Sadok, and J. Kelner, "An IoT sensor and scenario survey for data researchers," *Journal of the Brazilian Computer Society*, vol. 25, no. 1, p. 4, Dec. 2019/
19. J. J. Barriga et al., "Smart Parking: A Literature Review from the Technological Perspective," *Applied Sciences*, vol. 9, no. 21, p. 4569, Oct. 2019.
20. The OWASP Foundation, "OWASP Top Ten," OWASP Top Ten, Sep. 30, 2021.
21. P. Polleit and M. Spreitzenbarth, "Defeating the Secrets of OTP Apps," in 2018 11th International Conference on IT Security Incident Management & IT Forensics (IMF), May 2018, pp. 76–88.
22. S. Babkin and A. Epishkina, "Authentication Protocols Based on One-Time Passwords," in 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EConRus), Jan. 2019, pp. 1794–1798.
23. D. M'Raihi, M. Bellare, F. Hoornaert, D. Naccache, and O. Ranen, "HOTP: An HMAC-Based One-Time Password Algorithm," Dec. 2005. doi: 10.17487/rfc4226.
24. D. M'Raihi, S. Machani, M. Pei, and J. Rydell, "TOTP: Time-Based One-Time Password Algorithm," May 2011. doi: 10.17487/rfc6238.
25. B. B. Balilo, B. D. Gerardo, R. P. Medina, and Y. Byun, "Design of physical authentication based on OTP KeyPad," in 2017 International Conference on Applied Computer and Communication Technologies (ComCom), May 2017, pp. 1–5.
26. H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication," Feb. 1997. doi: 10.17487/rfc2104.
27. Lumburovska Lina, Dobрева Jovana, Andonov Stefan, Mihajloska Trpcheska, and Hristina Dimitrova Vesna, "A Comparative Analysis of HOTP and TOTP Authentication Algorithms. Which one to choose?" *International Scientific Journals of Scientific Technical Union of Mechanical Engineering "Industry 4.0,"* vol. 5, no. 4, pp. 131–136, 2021.
28. Lina Lumburovska, Jovana Dobрева, Stefan Andonov, Hristina Mihajloska Trpcheska, and Vesna Dimitrova, "A Comparative Analysis of HOTP and TOTP Authentication Algorithms. Which one to choose?" *International Scientific Journals of Scientific Technical Union of Mechanical Engineering "Industry 4.0,"* vol. 5, no. 4, pp. 131–136, 2021.
29. D. M'Raihi, M. Bellare, F. Hoornaert, D. Naccache, and O. Ranen, "HOTP: An HMAC-Based One-Time Password Algorithm," Dec. 2005. doi: 10.17487/rfc4226.
30. N. Haller, C. Metz, P. Nesser, and M. Straw, "A One-Time Password System," Feb. 1998. doi: 10.17487/rfc2289.
31. R. A. Grimes, "One-Time Password Attacks," in *Hacking Multifactor Authentication*, 2021, pp. 205–226. doi: 10.1002/9781119672357.ch9.
32. N. Nassar and L.-C. Chen, "Seed-based authentication," in 2015 International Conference on Collaboration Technologies and Systems (CTS), Jun. 2015, pp. 345–350.
33. S. Ma et al., "An empirical study of SMS one-time password authentication in Android apps," in *Proceedings of the 35th Annual Computer Security Applications Conference*, Dec. 2019, pp. 339–354.
34. C. Peeters, C. Patton, I. N. S. Munyaka, D. Olszewski, T. Shrimpton, and P. Traynor, "SMS OTP Security (SOS)," in *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, May 2022, pp. 2–16.
35. Md. H. Berenjestanaki, M. Conti, and A. Gangwal, "On the Exploitation of Online SMS Receiving Services to Forge ID Verification," in *Proceedings of the 14th International*

- Conference on Availability, Reliability and Security, Aug. 2019, pp. 1–5. doi: 10.1145/3339252.3339276.
36. Ananthi Sheshasaayee and D. Sumathy, “A Framework to Enhance Security for OTP SMS in E-Banking Environment Using Cryptography and Text Steganography,” 2017, pp. 709–717. doi: 10.1007/978-981-10-1678-3\_68.
  37. M. Imanullah and Y. Reswan, “Randomized QR-code scanning for a low-cost secured attendance system,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 4, p. 3762, Aug. 2022, doi: 10.11591/ijece.v12i4.pp3762-3769.
  38. K. Krombholz, P. Frühwirth, P. Kieseberg, I. Kapsalis, M. Huber, and E. Weippl, “QR Code Security: A Survey of Attacks and Challenges for Usable Security,” 2014, pp. 79–90.
  39. Y. Oren and D. Arad, “Toward Usable and Accessible Two-Factor Authentication Based on the Piezo-Gyro Channel,” *IEEE Access*, vol. 10, pp. 19551–19557, 2022.
  40. S. A. Lone and A. H. Mir, “A novel OTP based tripartite authentication scheme,” *International Journal of Pervasive Computing and Communications*, vol. 18, no. 4, pp. 437–459, Jul. 2022.
  41. Fortinet, “FortiToken One-Time Password Token,” <https://www.fortinet.com/content/dam/fortinet/assets/data-sheets/fortitoken.pdf>, Nov. 29, 2022.
  42. K. Aravindhana, “One-time Password: A Survey,” *International Journal of Emerging Trends in Engineering and Development Issue 3*, vol. 1, no. 3, 2013.
  43. Google, “Sign in with backup codes.” <https://support.google.com/accounts/answer/1187538?hl=en&co=GENIE.Platform%3DAndroid> (accessed Jan. 23, 2023).
  44. R. A. Grimes, “Types of Authentications,” in *Hacking Multifactor Authentication*, Wiley, 2020, pp. 59–99. doi: 10.1002/9781119672357.ch3.
  45. D. Tirfe and V. K. Anand, “A Survey on Trends of Two-Factor Authentication,” 2022, pp. 285–296. doi: 10.1007/978-981-16-4244-9\_23.
  46. M. A. Hassan and Z. Shukur, “Device Identity-Based User Authentication on Electronic Payment System for Secure E-Wallet Apps,” *Electronics (Basel)*, vol. 11, no. 1, p. 4, Dec. 2021, doi: 10.3390/electronics11010004.
  47. C. Sudar, S. K. Arjun, and L. R. Deepthi, “Time-based one-time password for Wi-Fi authentication and security,” in *2017 International Conference on Advances in Computing, Communications, and Informatics (ICACCI)*, Sep. 2017, pp. 1212–1216.
  48. V. Kameswaran and S. Hulikal Muralidhar, “Cash, Digital Payments and Accessibility,” *Proc ACM Hum Comput Interact*, vol. 3, no. CSCW, pp. 1–23, Nov. 2019.
  49. S. Singanamalla, V. Potluri, C. Scott, and I. Medhi-Thies, “PocketATM,” in *Proceedings of the Tenth International Conference on Information and Communication Technologies and Development*, Jan. 2019, pp. 1–11. doi: 10.1145/3287098.3287106.
  50. C. Stephens, “Why are SMS codes still the global ID solution?” *Biometric Technology Today*, vol. 2020, no. 8, pp. 8–10, Sep. 2020, doi: 10.1016/S0969-4765(20)30110-7.
  51. J. Kook, “Design and Implementation of a OTP-based IoT Digital Door-lock System and Applications,” 2019. [Online]. Available: <http://www.irphouse.com>
  52. T. Mahboob Alam et al., “OTP-Based Software-Defined Cloud Architecture for Secure Dynamic Routing,” *Computers, Materials & Continua*, vol. 71, no. 1, pp. 1035–1049, 2022.
  53. J.-N. Luo, C.-M. Wu, and M.-H. Yang, “A CAN-Bus Lightweight Authentication Scheme,” *Sensors*, vol. 21, no. 21, p. 7069, Oct. 2021, doi: 10.3390/s21217069.
  54. M. Gawas, H. Patil, and S. S. Govekar, “An integrative approach for secure data sharing in vehicular edge computing using Blockchain,” *Peer Peer Netw Appl*, vol. 14, no. 5, pp. 2840–2857, Sep. 2021, doi: 10.1007/s12083-021-01107-4.
  55. V. A. Cunha, D. Corujo, J. P. Barraca, and R. L. Aguiar, “TOTP Moving Target Defense for sensitive network services,” *Pervasive Mob Comput*, vol. 74, p. 101412, Jul. 2021.