

Czech University of Life Sciences Prague

Faculty of Economics and Management

Department of information Engineering



Bachelor Thesis

Authentication using one-time passwords

Autor: Timashev Roman

Supervisor: Ing. Josef Pavlicek, Ph.D

© 2015 CULS Prague

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Katedra informačního inženýrství

Provozně ekonomická fakulta

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Roman Timashev

Informatics

Název práce

One time login password

Název anglicky

One time login password

Cíle práce

The thesis is thematically focused on system of one-time passwords (OTP). The main goals are:

- 1) To show the basic principles of work one-time passwords.
- 2) To find the safest method of delivering OTP to the user;
- 3) To find the best method of generating the OTP;
- 4) Try to find the possible causes of of hacking one-time passwords and try to improve this solution according to user requirements and convenience;
- 5) To design and implement this solution.

Partial goals of this work are:

- 1) To analyze the general types of authentication;
- 2) To show the advantages and disadvantages of OTP system;
- 3) To show methods of delivering, generating of one-time passwords.

Metodika

Methodology of these objectives is based on study and analysis of specialized information resources.

In the theoretical part of thesis, we will describe the different types of authentication, methods of delivering OTPs, reasons of using these passwords and their types. In addition, we will consider the strengths and weaknesses of using OTP. We will describe the different methods of generating numbers for passwords using some different types of algorithms, the different type of tokens, which one is better for our solutions. In the end of this part, we will try to find the best algorithm of this system; the best delivering and generating methods of one-time passwords for web application. According to this knowledge, we will try to apply it in practice.

In the practical part of the work explains the principle of using OTP for web application using MultiOTP utility with PHP library. We will use different types software tokens on the Android system. In addition, I will try to improve the protection of using hardware tokens for internet banking to provide secure transaction.

Based on theoretical knowledge and the results of my solution, the conclusions of the thesis will be formulated.



Doporučený rozsah práce

35 str

Klíčová slova

One-time password, OTP, Authentication, HOTP, TOTP

Doporučené zdroje informací

KIM, Tai-Hoon, Hojjat ADELI, John ROBLES a Maricel BALITANAS. Advanced Communication and Networking: International Conference, ACN 2011, Brno, Czech Republic, August 15-17, 2011, Proceedings. Verlag Berlin Heidelberg: Springer, 2011. ISBN 978-3-642-23312-8.

LEHTINEN, Rick, Deborah RUSSELL, G GANGEMI a Deborah RUSSELL. Computer security basics. 2nd ed. Sebastopol, CA: O'Reilly, c2006, xii, 296 p. ISBN 05-960-0669-1.

Předběžný termín obhajoby

2015/06 (červen)

Vedoucí práce

Ing. Josef Pavlíček, Ph.D.

Elektronicky schváleno dne 31. 3. 2015

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 31. 03. 2015

Declaration

I declare that I have worked on my Bachelor thesis titled "**Authentication using one-time passwords**" by myself and I have used only the sources mentioned at the end of the thesis. As the author of the diploma thesis, I declare that the thesis does not break copyrights of any third person.

In Prague on 16.03.2015

Roman Timashev

Acknowledgement

I would like to thank Ing. Josef Pavlíček, Ph.D. for his advises and supervision of my bachelor thesis.

Autentizaci pomocí jednorázová hesla

Authentication using one-time passwords

Souhrn

V této bakalářské práci jsou vysvětleny základní principy a vylepšení ověření jednorázových hesel. Jméno jednorázových hesel mluví samo za sebe. Hesla tohoto typu lze použít pouze jednou v registraci nebo transakci. Na rozdíl od pravidelných hesel, jednorázová hesla je obtížné krást, protože již nejsou platné. Autentizace pomocí OTP - jeden z nejsilnějších ověřování v tuto chvíli.

Vybral jsem si touto téma, protože frekvence jí využití v našich dnech. Spousta různých společností a banku úspěšně používají tuto technologii na ochranu svých klientů z různých útoků. I když téma hesel není pro nás nova, ale většina velkých společností, jako je Google nebo Facebook používají autentikaci jednorázových hesel, protože oni vidí budoucnost v tomto systému.

Summary

In this Bachelor thesis are explained the basic principles and improvements of one-time password authentication. The name of one-time passwords speaks for itself. Passwords of this type can be used only once in the registration or transaction. As opposed to regular passwords, one-time passwords are difficult to steal because they no longer valid. Authentication using OTPs - one of the strongest authentication at the moment.

I chose this topic because of its frequency of use in our days. Lots of different companies and banks successfully apply this technology to protect their clients from various attacks. Although the topic of passwords is not new to us, but most of the large companies such as Google or Facebook use authentication with one-time passwords, because they see a future in this system.

Klíčová slova: Jednorázová hesla, OTP, Autentizace, HOTP, TOTP, Tokeny

Keywords: One-time password, OTP, Authentication, HOTP, TOTP, Tokens

Content

1.	Introduction	3
1.1	Objectives	3
1.2	Methodology	3
2.	Theoretical part	4
2.1	Autentification	4
2.2	Basic authentication	5
2.3	One time password	6
2.3.1	Advantages of OTP	7
2.3.2	Disadvantages of OTP	8
2.4	Other authentication methods	9
2.4.1	Biometrics	9
2.4.2	Asymmetric cryptograpy	10
2.5	Methods of generating the OTP	11
2.5.1	Time-based synchronization	11
2.5.2	Using mathematical algorithm	13
2.5.3	Challenge-Response (Asynchronous Authentication)	14
2.5.4	S/Key One-time password system	15
2.6	Methods of delivering of OTP	16
2.6.1	Text messaging	16
2.6.2	Web-based method	17
2.6.3	Hardware tokens	19
2.6.4	Software tokens	22
2.6.5	Information on the paper	23
2.7	Other systems of authentication	23
2.7.1	SecurID	23
2.7.2	Keep Your Password Secret	25
2.8	One OTP implementation versus another	25
3.	Practicle part: Authentication for web application using one-time passwords.	28
3.1	Registration for web application	30
3.2	Logon	31
3.3	Resynchronization existing users	32
3.4	Database file	33
3.5	Types of improving the protection	34
4.	Conclusion	37
5.	List of figures	38
6.	List of references:	39
5.1	List of books	39
5.2	List of electronic sources	40
7.	List of tables	41
8.	Annex: Part of the code	41

1. Introduction

1.1 Objectives

The thesis is thematically focused on system of one-time passwords (OTP).

The main goals are:

- 1) To show the basic principles of work one-time passwords.
- 2) To find the safest method of delivering OTP to the user;
- 3) To find the best method of generating the OTP;
- 4) Try to find the possible causes of of hacking one-time passwords and try to improve this solution according to user requirements and convenience;
- 5) To design and implement this solution.

Partial goals of this work are:

- 1) To analyze the general types of authentication;
- 2) To show the advantages and disadvantages of OTP system;
- 3) To show methods of delivering, generating of one-time passwords.

1.2 Methodology

Methodology of these objectives is based on study and analysis of specialized information resources.

In the theoretical part of thesis, we will describe the different types of authentication, methods of delivering OTPs, reasons of using these passwords and their types. In addition, we will consider the strengths and weaknesses of using OTP. We will describe the different methods of generating numbers for passwords using some different types of algorithms, the different type of tokens, which one is better for our solutions. In the end of this part, we will try to find the best algorithm of this system; the best delivering

and generating methods of one-time passwords for web application. According to this knowledge, we will try to apply it in practice.

In the practical part of the work explains the principle of using OTP for web application using MultiOTP utility with PHP library. We will use different types software tokens on the Android system. In addition, I will try to improve the protection of using hardware tokens for internet banking to provide secure transaction.

Based on theoretical knowledge and the results of my solution, the conclusions of the thesis will be formulated.

2. Theoretical part

2.1 Authentication

The information age has put billions of people vulnerable to acts of theft of personal details. One situation would be millions of users using social networking sites such as Facebook, My Space and Twitter to name a few. People use these sites voluntarily and place all sorts of information out there for everyone to see and use as they desire. All of a sudden we have been given the opportunity to connect with an audience that we would have never had access to before. This has its advantages however it has many disadvantages as well. The information just needs to pass through the wrong hands and can easily be used to our detriment. *[Magalhaes, 2011]*

That is why the authentication is important part of information system.

Authentication is the process of validating user identity. The fact that the user claims to be represented by a specific abstract object (identified by its user ID) does not necessarily mean that this is true. *[Todorov, 2007]*

We have three ways to authenticate a person, this is based on:

- What the person has (something physical);
- What the person knows (something private like a password or PIN¹);
- What the person is (Biometrics). *[Rouit, 2013]*

If we use two or more completely independent authentication methods (combination of these two groups above) we can talk about two factor authentication (2FA) or multi-factor authentication.

2FA is a security beyond passwords. The time of securing your data by password (single factor authentication) is no longer a viable choice; it is no longer rigorous enough to have as the sole security method. 2FA is a much more secure option. *[Magalhaes, 2011]*

2.2 Basic authentication

Basic Authentication was defined in HTML 1.0, so it has been around for a while and is the most widely supported form of authentication. Users are prompted for their credentials (a username and password), and those credentials are then encoded into a base64² string and sent back to IIS where they can be authenticated.

The biggest issue with Basic Authentication is that the username and password are sent over what amounts to clear text. Encoding and decoding a base64 string is very simple, so anyone with a network scanner can easily capture the encoded string, decode it, and determine someone's username and password. So, Basic Authentication isn't a secure form of authentication in and of itself. There is, of course, a caveat to that. Secure Sockets Layer (SSL) can encrypt all communications between a browser and IIS, including the base64 encoded string used in Basic Authentication.

Basic Authentication is a good option when compatibility is an issue because most browsers do support Basic Authentication. Basic Authentication also supports delegation, so it's useful in scenarios where you need to use delegation to access certain network resources. You should always be aware that Basic Authentication is inherently insecure and strive to secure the communication line between the browser and the server when using it. *[Armstrong, 2005]*

¹Personal identification Number

² Base64 is an encoding and decoding technique used to convert binary data to an American Standard

2.3 One time password

An OTP is password that is valid for only one login session or transaction. OTPs avoid a number of shortcomings that are associated with traditional passwords.

A one time password system uses a different password every time you want to authenticate yourself. Each password is used only once; thus, the term “one-time”.

The most important shortcoming that is addressed by OTPs is that, in contrast to static passwords, they are not vulnerable to replay attacks. This means that, if a potential intruder manages to record an OTP that was already used to log into a service or to conduct a transaction; he or she will not be able to abuse it since it will be no longer valid. On the downside, OTPs are difficult for human beings to memorize. Therefore they require additional technology in order to work. *[Roebuck, 2012]*

There is two main standard for generating One-Time Passwords: **HOTP** and **TOTP**, both of which are governed by the Initiative For Open Authentication (OATH). In the followings we will discuss the differences between these algorithms and finally we will attempt to use these authentication mechanisms with OpenAM.

- Hmac-based One-Time Password algorithm

This algorithm relies on two basic things: a shared secret and a moving factor (a.k.a counter). As part of the algorithm an HmacSHA1 hash (to be precise it's a hash-based message authentication code) of the moving factor will be generated using the shared secret. This algorithm is event-based, meaning that whenever a new OTP is generated, the moving factor will be incremented, hence the subsequently generated passwords should be different each time.

- Time-based One-Time Password algorithm

This algorithm works similarly to HOTP: it also relies on a shared secret and a moving factor, however the moving factor works a bit differently. In case of TOTP, the moving factor constantly changes based on the time passed since an epoch. The HmacSHA1 is calculated in the same way as with HOTP.

Why is the OTP a very strong authentication method?

There are few reasons why this is a very strong method:

- The key is 20 digits
- A password is a couple counter/password, only valid once and a very short time
- The algorithm that generates each password is not reversible
- With an OTP token, the key is hardware protected
- If the OTP is received on your phone, the key always stays at the server

Those few characteristics make the OTP a strong authentication protocol. The weakness in an authentication is usually the human factor. It is difficult to remember many complex passwords, so users often use the same one all across the internet and not really a strong one. With an OTP, you don't have to remember a password, the most you would have to remember would be PIN code (4 to 8 digits) if the OTP token is PIN protected. In the case of an OTP sent by a mobile phone, it is protected by your phone security. A PIN is short but you can't generally try it more than 3 times before the token is locked. *[Major, 2014]*

One-time password systems are being deployed by banks, governments, and corporate virtual private networks (VPNs) to reduce the damage of passwords compromised through phishing and some spyware attacks.

2.3.1 Advantages of OTP

One-time password systems can be easy to deploy and may not require any special software to be installed on the customer's computer. Some systems use one-time passwords generated on a hardware device that is communicated directly to the computer, say through a USB port. This option requires software to be installed (*Fig. 1*).

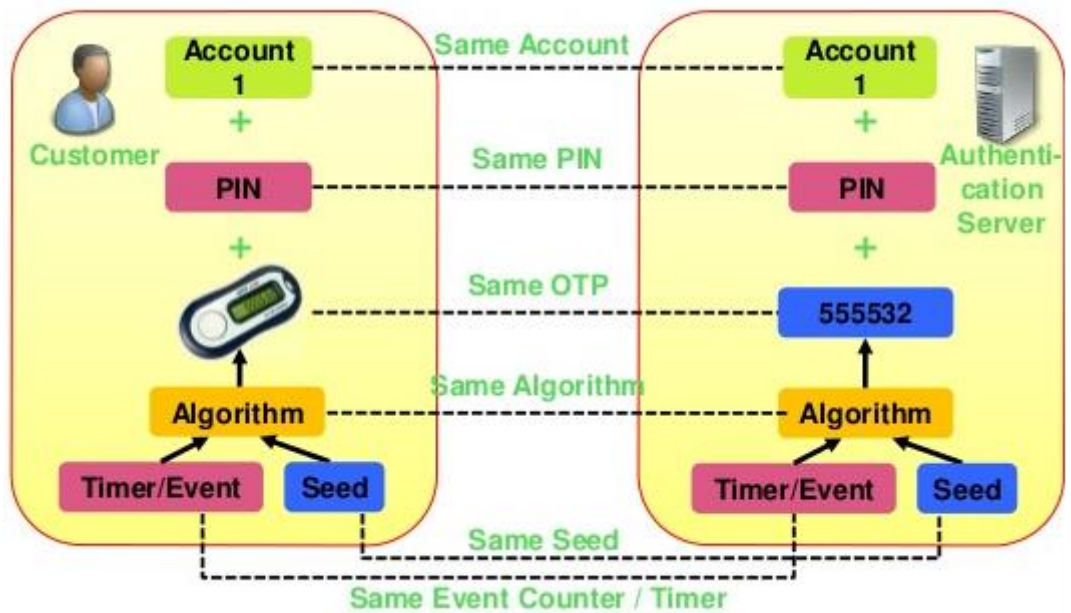


Figure 1 - Time synchronized OTP

source: www.slideshare.net

One-time password systems are generally acceptable to customers, due to their similarity to password systems. One-time password clock-based devices and challenge/response systems can be used across multiple systems (whereas counter-based solutions cannot without complicated re-synchronization). It is necessary that these are trusted systems, as each has the capability to impersonate the customer to the others. In practice, clock-based systems may also require time synchronization to work effectively.

With hardware one-time password devices and printed lists, the customer is likely to notice the loss if they are stolen. [Chaki, 2013]

2.3.2 Disadvantages of OTP

The verifier will need special software and/or hardware. Protected storage and management of the base secrets is required.

A disadvantage with clock-based one time passwords used across multiple systems is that there is a window of exposure: when a one-time password is used it can

be used with any of the other systems if an attacker obtains it. Shorter windows reduce the scope of such attacks. Also, these attacks may be countered by protecting the communication channel. Most hardware one-time password devices do not provide the same level of tamper resistance, and thus protection for the base secret, as hardware tokens do. This may change in the future as the hardware one-time password device market matures.

Systems based on shared printed tables, sometimes called bingo cards, have the same problems as written-down passwords: they may be copied or discovered and used without the customer's knowledge. Loss of the authentication key itself is a much more severe breach of security than the loss of any single one-time password. Shared tables exist that conceal the numbers under a coating, called scratchy cards, with the customer removing the coating to reveal each one-time password. These cards defend against copying attacks. They may still be stolen and used, although the customer would be expected to notice the loss of their card.

With authentication key sharing, the extent of the problem here would relate to how easy it is to copy. If copying is easy, then the customer can share their authentication key without losing the ability to authenticate. If copying is not feasible, then this may deter customers from sharing their authentication key, as they must also give up their ability to authenticate. *[Chaki, 2013]*

2.4 Other authentication methods

2.4.1 Biometrics

Identification by physical characteristics is as old as humanity. Recognizing people by their voices or appearance, and impersonating people by assuming their appearance, was widely known in classical times. Efforts to find physical characteristics that uniquely identify people include the Bertillion cranial maps, fingerprints, and DNA sampling. Using such a feature to identify people for a computer would ideally eliminate errors in authentication.

Biometrics is the automated measurement of biological or behavioral features that identify a person. When a user is given an account, the system administration takes a set of measurements that identify that user to an acceptable degree of error. Whenever the user accesses the system, the biometric authentication mechanism verifies the identity. Lawton points out that this is considerably easier than identifying the user because no searching is required. A comparison to the known data for the claimed user's identity will either verify or reject the claim. Common characteristics are fingerprints, voice characteristics, eyes, facial features, and keystroke dynamics.[Bishop, 2003]

2.4.2 Asymmetric cryptography

Asymmetric cryptography is also called public key cryptography. It uses key pairs consisting of a public key and a private key (Fig. 2). Each communication partner in an asymmetric cryptography solution needs their own unique key pair set (i.e. a private key and a public key); this makes asymmetric cryptography much more scalable than symmetric. The private key of the key pair must be kept private and secure. The public key of the key pair is distributed freely and openly. [Stewart, 2004]

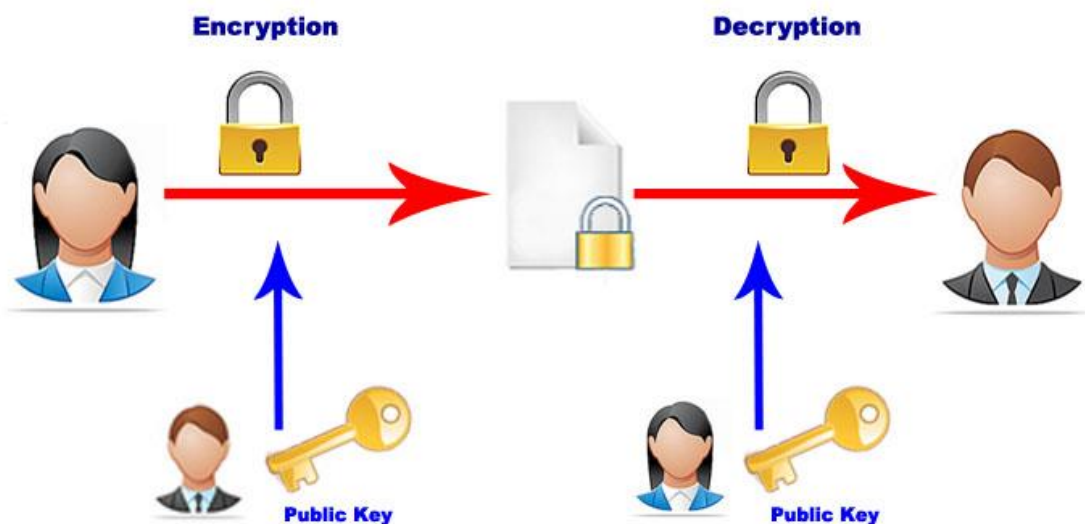


Figure 2 - An asymmetric encryption system

source: kryptophone.kryptotel.net

2.5 Methods of generating the OTP

OTP generation algorithms typically make use of randomness. This is necessary because otherwise it would be easy to predict future OTPs from observing previous ones. Concrete OTP algorithms vary greatly in their details. Various approaches for the generation of OTPs are listed below:

- Based on time-synchronization between the authentication server and the client providing the password (OTPs are valid only for a short period of time)
- Using a mathematical algorithm to generate a new password based on the previous password (OTPs are effectively a chain and must be used in a predefined order).
- Using a mathematical algorithm where the new password is based on a challenge (e.g., a random number chosen by the authentication server or transaction details) and/or a counter.

There are also different ways to make the user aware of the next OTP to use. Some systems use special electronic tokens that the user carries and that generate OTPs and show them using a small display. Other systems consist of software that runs on the user's mobile phone. Yet other systems generate OTPs on the server-side and send them to the user using an out-of-band channel such as SMS messaging. Finally, in some systems, OTPs are printed on paper that the user is required to carry with them. [Roebuck, 2012]

2.5.1 Time-based synchronization

Time-only, synchronous authentication is based on time clocks and secret keys that reside in two places: on the network (i.e., protected) side and on the user side (i.e., the side to be authenticated). On the network side, a time clock and database of secret keys operate in either a dedicated authentication hardware box or in a software authentication server. On the user side of the authentication equation, a clock, which is synchronized to the authentication server, and a secret key (corresponding to a secret key in the server) operate inside the token (*fig.3*).

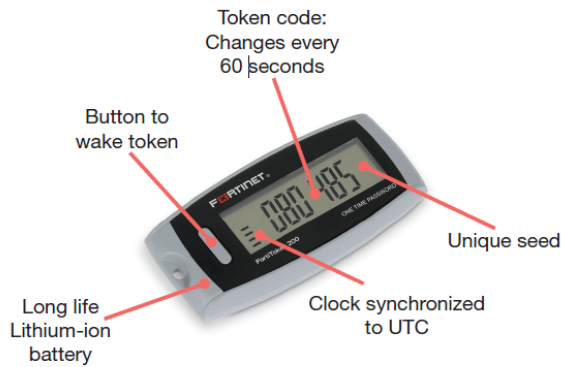
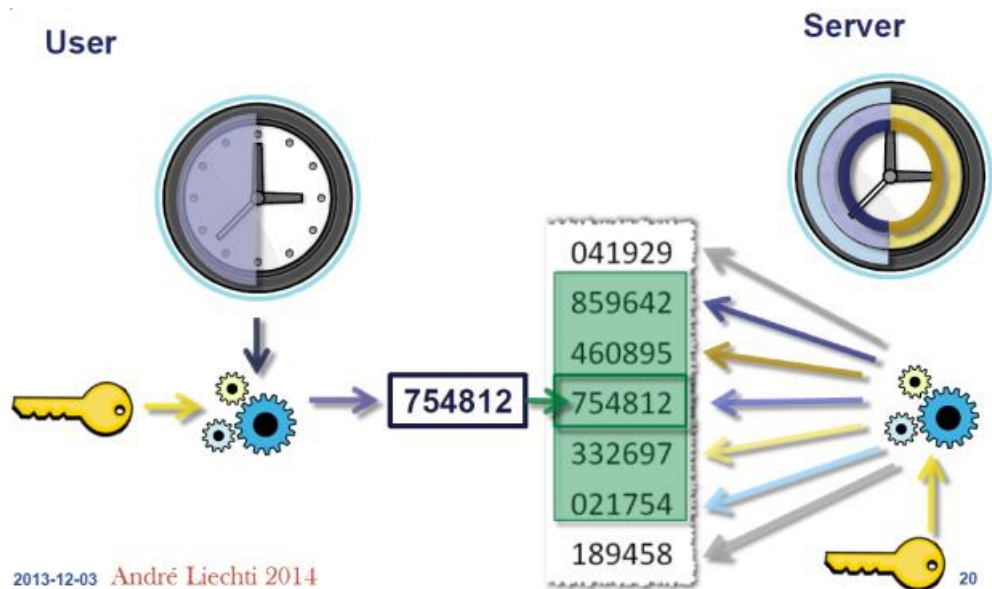


Figure 3 - Time Based e-Token

source: www.avfirewalls.com



2013-12-03 André Liechti 2014

Figure 4 - Time-synchronous scheme

source: cybermashup.com/

Several implementations are possible of time-only, synchronous authentication. In one specific, time-synchronous scheme (*fig. 4*), a proprietary algorithm continually executes in the token to generate access codes based on the time clock and the secret key of the token. In this case, the time is the “variable.” A new access code is generated by the token approximately once a minute. The token is always activated. When the user dials in to the authentication server, the server issues a prompt to the user for an access code. The user simply attaches his or her secret Personal identification Number (PIN) to the code currently displayed on his token at the moment access is required, and then the user transmits the combined PIN and code (which become the “one-time password”). This code is transmitted over telephone lines to the authentication server. The server

uses the PIN to identify the user to compare the transmitted access code with its own current version for that user. [Slone, 1999]

2.5.2 Using mathematical algorithm

Lamport proposed a one-time password scheme (Lamport, 1981) that can be implemented without special hardware. Assume there is some function F that is reasonably easy to compute in the forward direction but effectively impossible to invert. Further assume that the user has some secret—perhaps a password— $\langle S \rangle$. To enable the user to log in some number of times, the host calculates $F(s)$ that number of times. Thus, to allow 1000 logins before a password change, the host would calculate $F^{1000}(s)$, and store only that value. [Cheswick, 2003]

Using a mathematical algorithm where the new password is based on a challenge (e.g., a random number chosen by the authentication server or transaction details) and/or a counter. Each new OTP may be created from the past OTPs used. An example of this type of algorithm, credited to Leslie Lamport, uses a one-way function (call it f). The one-time password system works by starting with an initial seed s , then generating passwords $f(s)$, $f(f(s))$, $f(f(f(s)))$, ... as many times as necessary. If an indefinite series of passwords is wanted, a new seed value can be chosen after the set for s is exhausted. Each password is then dispensed in reverse, with $f(f(\dots f(s)))$ first, to $f(s)$.

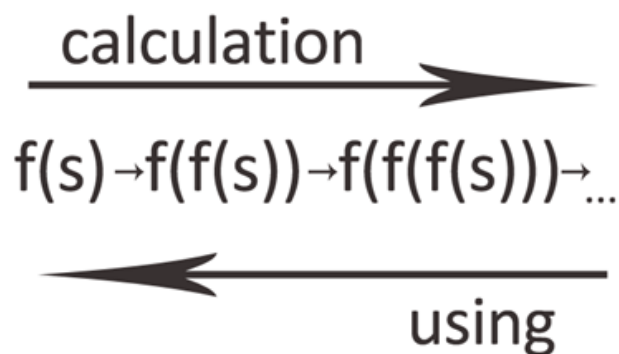


Figure 5 - algorithm for generating numbers

source: autor

If an intruder happens to see a one-time password, he may have access for one time period or login, but it becomes useless once that period expires. To get the next password in the series from the previous passwords, one needs to find a way of calculating the inverse function f^{-1} . Since f was chosen to be one-way, this is extremely difficult to do. If f is a cryptographic hash function, which is generally the case, it is (so far as is known) a computationally infeasible task. In some mathematical algorithm schemes, it is possible for the user to provide the server with a static key for use as an encryption key, by only sending a one-time password. [Chaki, 2013]

2.5.3 Challenge-Response (Asynchronous Authentication)

The use of challenge-response one-time passwords will require a user to provide a response to a challenge. For example, this can be done by inputting the value that the token has generated into the token itself. To avoid duplicates, an additional counter is usually involved, so if one happens to get the same challenge twice, this still results in different one-time passwords. However, the computation does not usually involve the previous one-time password; i.e. usually this or another algorithm is used, rather than using both algorithms. [Chaki, 2013]

```
challenge: 00193 Wed Sep 11 11:22:09 2002
response:  ab0dh1kd0jkfj1kye./
```

Figure 6 - an example of asynchronous authentication

source: autor

No one except the network manager or administrator has access to the database of user secret keys and other pertinent user information. A LAN dial-up remote access can provide an example on how this works. A user dials up remotely, and before the network allows the user access, the call is intercepted by a master authentication device (or a software authentication server), which prompts the user for an ID. When the user is identified as one of the individuals allowed access to the network, the server issues a random alphanumeric challenge to begin the process of authenticating (i.e. determining that the user is who he says he is).

That random challenge is used by both the token and the server to calculate a one-time-use password based on a secret key value stored in both the token and the server. The reliability of the algorithm used in the authentication solution of an organization should be carefully evaluated.

Solutions that employ the challenge-response process, secret user keys, and encryption algorithms to generate passwords result in a very high level. [Slone, 1999]

2.5.4 S/Key One-time password system

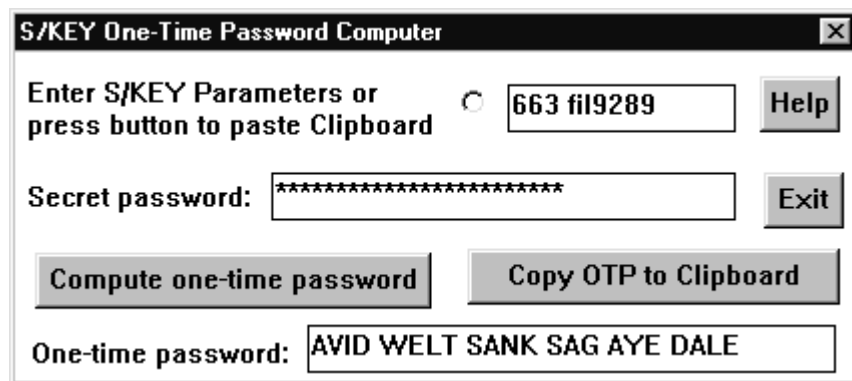


Figure 7 - S/KEY OTP generator example

source: techpubs.sgi.com

As mentioned earlier, S/KEY is a seminal OTP system developed for authentication at Bellcore. Using this system, the real password is never transmitted across the network. Instead, the real password is combined with a short set of characters and a decrementing counter to form a new single-use, one-time password. The S/KEY OTP system generates a password based on a seed secret pass phrase with a secure hash function such as MD5. The S/KEY server verifies the one-time password by making a pass through the secure hash algorithm and comparing the result with the previous password.

Inverting the hash function that produced the one-time single-use password is extremely difficult. However, S/KEY is sensitive to man-in-the-middle attacks. A secure transport layer protocol (SSL/TLS) can be used to counteract this. S/KEY one-time password is documented in IETF RFC I760. [Bhaiji, 2008]

2.6 Methods of delivering of OTP

2.6.1 Text messaging

A common technology used for the delivery of OTPs is text messaging. Because text messaging is a ubiquitous communication channel, being available in nearly all handsets and with a large customer-base, text messaging has a great potential to reach all consumers with a low total cost to implement. However, the cost of each text messaging often for each OTP might not be suitable for some users. OTP over text messaging may also be encrypted using an A5/X standard which several hacking groups report can be successfully decrypted within minutes or seconds or the OTP over SMS might not be encrypted by one's service-provider at all. In addition to threats from hackers, the mobile phone operator becomes part of the trust chain. In the case of roaming, more than a single mobile phone operator has to be trusted. Anyone using this information may mount a man-in-the-middle attack. *[Roebuck, 2012]*

Using SMS text messages as a conduit to assure user identity presents several security concerns:

- SMS Spoofing and number redirection
- Phone cloning and hijacking
- Multiple OTP (One-Time-Password) messages & day codes present a significant window of opportunity for intrusion
- Even if the phone has a lock code it may still display text messages when in a locked mode

Users may be authenticated using one of three established authentication factors: Something the user knows, something the user has and something the user is. Combinations of two or more of these factors produce strong authentication to safeguard both the user and the service provider. *[Sullivan, 2013]*

Facebook has added a one-time password feature as part of an effort to address account security.

The social network site is gradually rolling out the ability to have Facebook text a one-time password to users concerned about working on machines other than their normal computers, such as public computers in hotels, cafes or airports.

"Simply text 'otp' to 32665 on your mobile phone, and you'll immediately receive a password that can be used only once and expires in 20 minutes". [Prince, 2010]



Figure 8 - Received message with OTP via SMS

source: www.itrustsecurity.com

One-time password via SMS is often used in the internet banking. The user enters their data to your account, and always before the transaction receive authorization SMS with a code that must be entered into the form.

2.6.2 Web-based method

Authentication-as-a-service providers offer various web-based methods for delivering one-time passwords without the need for tokens.



The pictures, their location, and the alphanumeric characters are different every time, but the user always looks for their same categories.

Figure 9 - Randomly generated grid of pictures

source: confidenttechnologies.com

One such method relies on the user's ability to recognize pre-chosen categories from a randomly-generated grid of pictures. When first registering on a website, the user chooses several secret categories of things; such as dogs, cars, boats and flowers. Each time the user logs into the website they are presented with a randomly-generated grid of pictures. Each picture in the grid has a randomly-generated alphanumeric character overlaid on it. The user looks for the pictures that fit their pre-chosen categories and enters the associated alphanumeric characters to form a one-time access code. [Roebuck, 2012]

This method is similar with OTP method. The image title set by a user can be a secret of each user and different images are differently arranged and different access codes are assigned each time so it can be said that it is a kind of OTP method. However, if a user chooses 3 different image titles and 9 image samples are offered, only 6 out of 504 combination codes ($9 \times 8 \times 7$) can be suitable access codes, which means 1.19% probability. This method has a problem that three image categories can be easily found out through the analysis of similarities and commonness of 9 images that are given when an attacker tries to have access several times. However, in that a one-time access code is generated by use of image information, this method shows that it is

possible to connect the existing general OTP method with images or other multimedia information. Therefore, this study intends to suggest a user authentication process that each user generates a one-time password from a random image captured by the user through his/her smart phone, and performs user authentication. [Kim, 2011]

2.6.3 Hardware tokens



Figure 12 - Token with biometric data
source: i00.i.aliimg.com



Figure 11 - Waterproof token
source: verisign.com



Figure 10 - Credit card token
source: surepassid.com

A security token may be a physical device that an authorized user of computer services is given to ease authentication.

Security tokens are used to prove one's identity electronically (as in the case of a customer trying to access their bank account). The token is used in addition to or in place of a password to prove that the customer is who they claim to be. The token acts like an electronic key to access something.

Some may store cryptographic keys, such as a digital signature, or biometric data, such as fingerprint minutiae (fig.12). Some designs feature tamper resistant packaging (fig.11), while others may include small keypads to allow entry of a PIN or a simple button to start a generating routine with some display capability to show a generated key number (fig.10). Special designs include a USB connector, RFID functions or Bluetooth wireless interface to enable transfer of a generated key number sequence to a client system. [Roebuck, 2012]



Figure 13 - Usb tokens

source: mcaert.safescrypt.com

One type of a token is a credit card-size device with a built-in keypad . At login, the server issues a challenge with a number. The user keys this number into the token card, and the card displays a response. The user inputs this response and sends it to the server, which calculates the same result it expects to see from the token. If the numbers match, the user is authenticated.

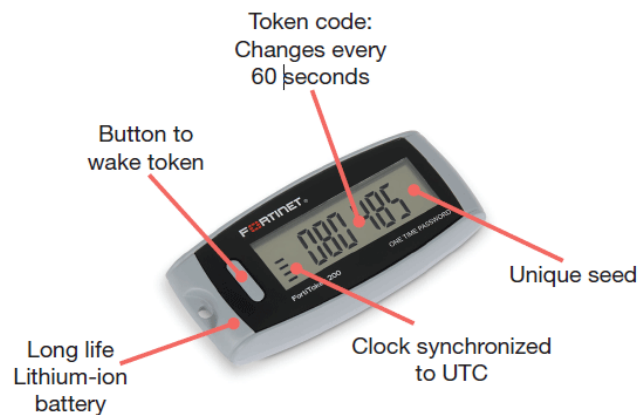


Figure 14 - Token based on time

source: avfirewalls.com

Other tokens are based on time (*fig. 12*). They display a number that changes at regular intervals, usually several times each hour. The user logs in by entering her username and password, along with the time-based value from the token. If the value from the token matches a value the server has calculated, the account is authenticated, and access is granted.

The disadvantage of tokens is their small size and their price. If the token breaks or becomes lost, a replacement will be needed to gain access. Depending on the

manufacturer and quantity purchased, tokens can run from S30 to S100 each. Software-based tokens are also available. At login, users enter a personal identification number (PIN), with which the soft token creates a one-time password. The PIN is never transmitted. The software token protects itself by refusing to work if an incorrect PIN is entered too many times. [Lehtinen, 2006]

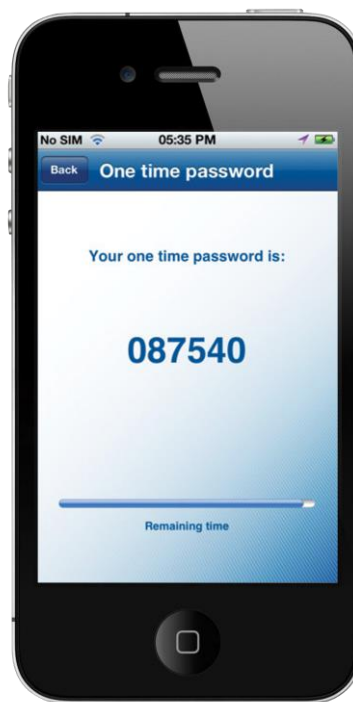


Figure 15 - Token on mobile phone

source: *vasco.com*

Alternatively the new form of tokens that are coming into main stream are mobile device which are communicated with out-of-band channel (like voice. SMS. USSD) that also make the authentication and identity protection much stronger when compared to conventional simple synchronous dynamic password tokens. [Roebuck, 2012]

The most simple vulnerability with any password containers is just losing the special key device or the activated smart phone with the integrated key function. Such vulnerability cannot be healed with any single token container device within the pre-set time span of activation. All further consideration presumes performant loss prevention, e.g. by additional electronic leash or body sensor and alarm. [Roebuck, 2012]

2.6.4 Software tokens

A software token is a type of two-factor authentication security device that may be used to authorize the use of computer services. Software tokens are stored on a general-purpose electronic device such as a desktop computer, laptop, PDA, or mobile phone. This is in contrast to hardware tokens, where the credentials are stored on a dedicated hardware device.

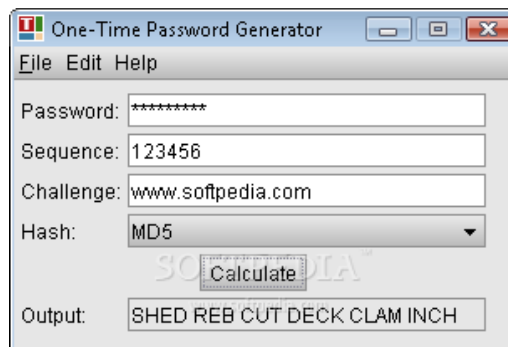


Figure 16 - OTP generator installed on PC

source: *softpedia.com*

Because software tokens are something one is not physically in possession of, they are exposed to unique threats such as computer viruses and software attacks. However, both hardware and software tokens are vulnerable to hot-based man-in-the-middle attacks, or simple phishing attacks in which the OTP provided by the token is solicited, and then supplied to the genuine website in a timely manner. Software tokens do have unarguable benefits: there is no physical token to carry, they do not contain batteries that will run out, and they are cheaper than hardware tokens. [Roebuck, 2012]

2.6.5 Information on the paper

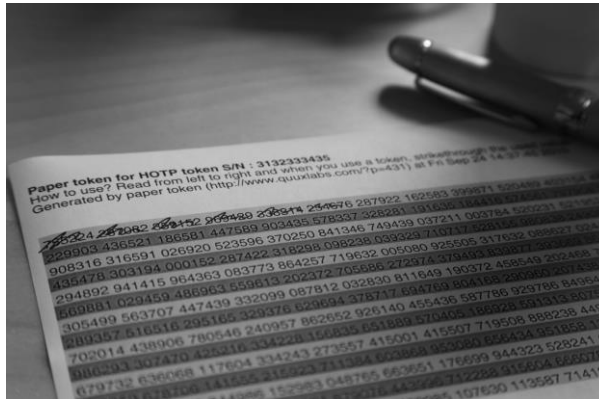


Figure 17 - Paper token

source: quuxlabs.com

In some countries online banking, the bank sends to the user a numbered list of OTPs that are printed on paper. For every online transaction, the user is required to enter a specific OTP from that list. In Germany, those OTPs are typically called TANs (for ‘transaction authentication numbers’). Some banks even dispatch such TANs to the user’s mobile phone via SMS, in which case they are called mTANs (for ‘mobile TANs’). [Roebuck, 2012]

2.7 Other systems of authentication





2.7.1 SecurID

RSA SecurID provides world-leading two-factor authentication, protecting 25,000 organizations and 55 million users. RSA SecurID extends security to bring your own device (BYOD), cloud, and mobile as well as traditional virtual private network (VPN) and web portals. RSA SecurID solutions comprise three primary components: authenticator, platform, and agent. [emc.com]

RSA SecurID is a one-time password scheme and an associated authentication mechanism developed by RSA Security. SecurID requires hardware that requires the use of tokens for user authentication, as well as a user PIN, and is therefore a two-factor authentication mechanism. RSA SecurID is a very popular authentication scheme used by many infrastructure solutions. [Todorov, 2007]

The RSA SecurID authentication mechanism consists of a "token"—a piece of hardware (e.g. a token or USB) or software (e.g. a "soft token" for a computer, PDA or cell phone) - assigned to a computer user that generates an authentication code at fixed intervals (usually 30 or 60 seconds) using a built-in clock and the card's factory-encoded random key (known as the "seed" and often provided as an ASCII file). The seed is different for each token, and is loaded into the corresponding RSA SecurID server (RSA Authentication Manager, formerly ACE/Server) as the tokens are purchased. The seed is typically 128 bits long. Some RSA SecurID deployments may use varied second rotations, such as 30-second increments. The token hardware is designed to be tamper-resistant to deter reverse engineering. Despite this, public code has been developed by the security community allowing a user to emulate RSA SecurID in software, but only if they have access to a current RSA SecurID code, and the original RSA SecurID seed file introduced to the server. [Roebuck, 2012]

Table 1 - Comparison of RSA hardware tokens **source: emc.com**

	Style	Available on	Artwork customization available	Number of digits displayed	Display change time (OTP)	Duration availability	Applications
RSA SecurID 200 Authenticator 	Card style	RSA Authentication Manager and SecurID Authentication Engine (SAE)	Yes	6 standard; can be customized to 4 or 8	60 seconds; can be customized to 30 seconds	24, 36, and 48 months	OTP
RSA SecurID 520 Authenticator 	Card style	RSA Authentication Manager and SecurID Authentication Engine (SAE)	Yes	6 standard; can be customized to 4 or 8	60 seconds; can be customized to 30 seconds	24, 36, and 48 months	Pin entry OTP
RSA SecurID 700 Authenticator 	Key fob	RSA Authentication Manager and SecurID Authentication Engine (SAE)	Yes	6 standard; can be customized to 4	60 seconds; can be customized to 30 seconds	24, 36, 48, and 60 months	OTP
RSA SecurID 800 Hybrid Authenticator 	Key fob	RSA Authentication Manager and SecurID Authentication Engine (SAE)	Yes	6 standard; can be customized to 4	60 seconds; can be customized to 30 seconds	24, 36, 48, and 60 months	OTP, email signing, HD/file encryption, etc.

2.7.2 Keep Your Password Secret

KYPS (Keep Your Password Secret) is a free web-based service that enables users to log into websites, which usually require a username/password combination, using one-time passwords. The main difference between KYPS and similar password management technologies is that the password is never disclosed to the local computer. This makes KYPS effective against password theft by spyware or keyloggers, particularly when using public computers such as in an internet cafe.

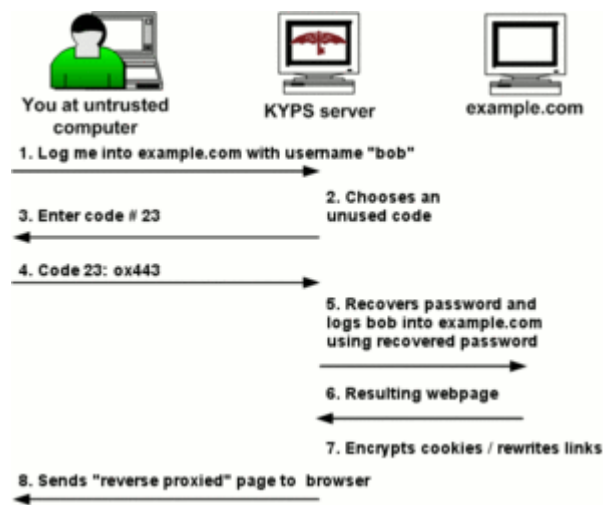


Figure 18- A sketch of the KYPS login method

source: en.wikipedia.org

The following figure sketches the login method of KYPS. Note that the user's password is not disclosed to the computer he uses. It is, however, disclosed to the KYPS server. The system therefore requires that the user trusts the provider of the service. [Roebuck, 2012]

2.8 One OTP implementation versus another

In terms of costs, the cheapest OTP solutions are those that deliver OTPs on paper, and those that generate OTPs on a device that someone already owns. This is because these systems avoid the costs associated with (re-)issuing proprietary electronic tokens and the cost of SMS messaging. For systems that rely on electronic tokens, algorithm-based OTP generators must cope with the situation where a token drifts out-of-sync with its server if the system requires the OTP to be entered by a deadline. This leads to an additional development cost. Time-synchronized systems, on the other hand, avoid this at the expense of having to maintain a clock in the electronic tokens (and an offset

value to account for clock drift). Whether or not OTPs are time-synchronized is basically irrelevant for the degree of vulnerability, it but avoids a need to reenter passwords if the server is expecting the last or next code that the token should be having because the server and token have drifted out-of-sync. Compared to most proprietary hardware tokens, so long as one already carries a phone or another mobile device in one's pocket, users of mobile devices don't need to carry and protect an extra item (which has no usefulness except that it generates OTPs). In addition to reducing costs considerably, using a phone as a token offers the convenience that it is not necessary to deliver devices to each end-user (who typically already own the device). For many users, a mobile phone may also be trickle-charged to preserve its battery for at least some portion of each day, whereas most proprietary tokens cannot be trickle-charged. However, most proprietary tokens have tamper-proof features. *[Roebuck, 2012]*

Certain conclusions

As we can see there are many options for the use of one-time passwords for different purposes. At first, compare the type of algorithms of OTPs. So we have a two different methods: HOTP and TOTP. Which one is better?

The main difference between HOTP and TOTP is that the HOTP passwords can be valid for an unknown amount of time, while the TOTP passwords keep on changing and are only valid for a short window in time. Because of this difference generally speaking the TOTP is considered as a more secure One-Time Password solution. *[Major, 2014]*

In case of internet banking we need to choose TOTP, because this method is more reliable for providing transactions on the Internet, but in the case authentication in the social network may cause inconvenience. HOTP can be used for protection in social networks, forums and any other web applications, that do not contain valuable information.

If we look at the methods of delivering of one-time passwords, we will see that the best of them are tokens. Obviously, many Internet banks use them as data protection. Hardware tokens with the best protection are tokens based on time. Of course, some banks still use the system to send messages via SMS, or just on paper and of course this

is extremely unsafe, an attacker can easily get your phone, or hack your mail. On mobile phone can be installed *malware*, which can send all data to attacker.

In February 2011 in Poland clients of ING Bank entered their phone number and mobile phones, then they got sms with a link to malware, which after installation all the data sent hacker. However, the authentication of mobile phone can not be called «weak». A lot of social networks, forums, banks still consider authentication with the mobile phone one of the most secure. Is caused by the fact that nowadays smart phones have become much more secure, such as Iphone 6 contains a fingerprint scanner, which is one of the type of human biometric data, without which is not as easy or impossible to obtain information on the phone.

Hardware tokens also have their drawbacks. It's a physical thing, it can also be lost or stolen, but if they are equipped with some protection, such as the user's biometric data (voice, fingerprint), then they are no longer of interest to the attacker, because the attacker has no private key, that is stored on the server. This degree of protection can be very effective for storing information.

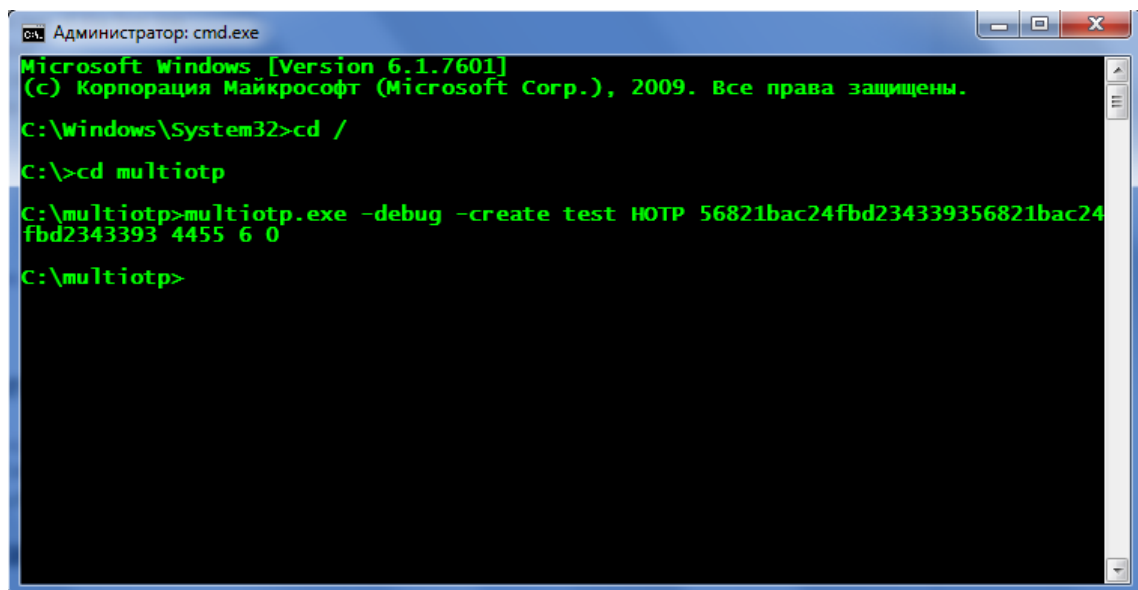
3. Practicle part: Authentication for web application using one-time passwords.

In this part of thesis we will try to use one-time passwords to log in web application. We will use MultiOTP, which is PHP class utility developed by SysCo to provide strong two factors authentication for applications. This utility supports software and hardware tokens with different algorithms (HOTP, TOTP and mobile-OTP).

MultiOTP also supports different types of servers such as RADUS server, Mongoose Web Server or directly PHP library using any web browser.

1. Downloading **multiotp.zip** package on your hard-drive from official web site.
2. Extract all files including multiotp.exe from the downloaded archive
3. Using the windows command line windows we should create user, which use HOTP algorithm
4. The following command will contain this structure:

multiotp.exe -debug -create <Name of user> Method of algorithm (in our case HOTP) <Some key> <Some PIN> <Length of token number> <livetime (in seconds)>



```
Администратор: cmd.exe
Microsoft Windows [Version 6.1.7601]
(c) Корпорация Майкрософт (Microsoft Corp.), 2009. Все права защищены.
C:\Windows\System32>cd /
C:\>cd multiotp
C:\multiotp>multiotp.exe -debug -create test HOTP 56821bac24fbd234339356821bac24
fbd2343393 4455 6 0
C:\multiotp>
```

Figure 19 - Creating user in system

source: autor

Hexadecimal -> base32 string coder

Hex string:

56821bac24fbd234339356821bac24fbd2343393

Note: all characters outside hex set will be ignored, thus "12AB34" = "12 AB 34" = "12, AB, 34", etc. Input is case-insensitive.

Options:

remove "0x" groups from input

Cleaned input:

56821BAC24FBD234339356821BAC24FBD2343393

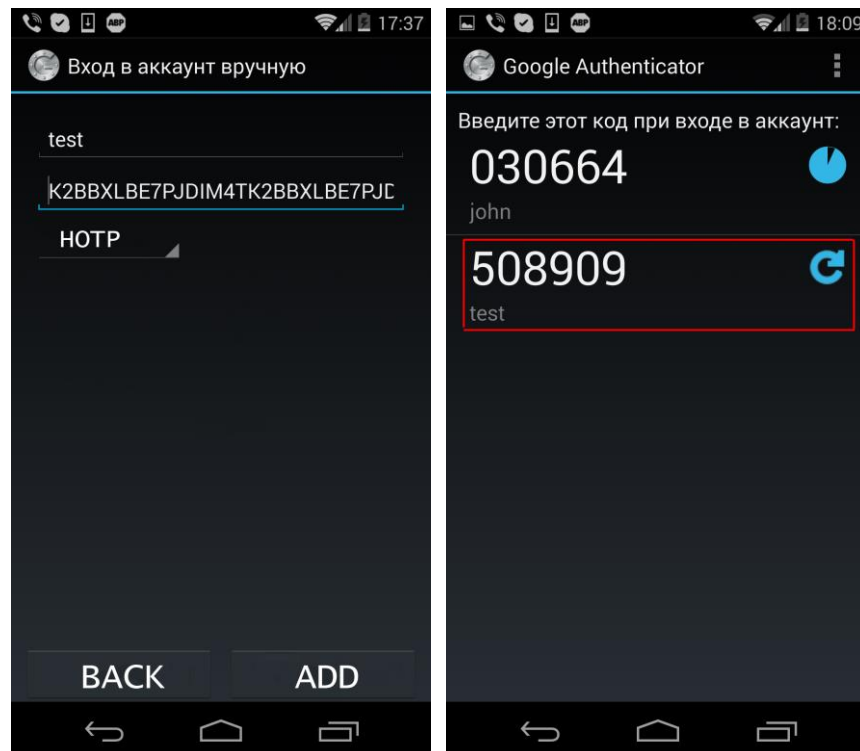
Output (base32):

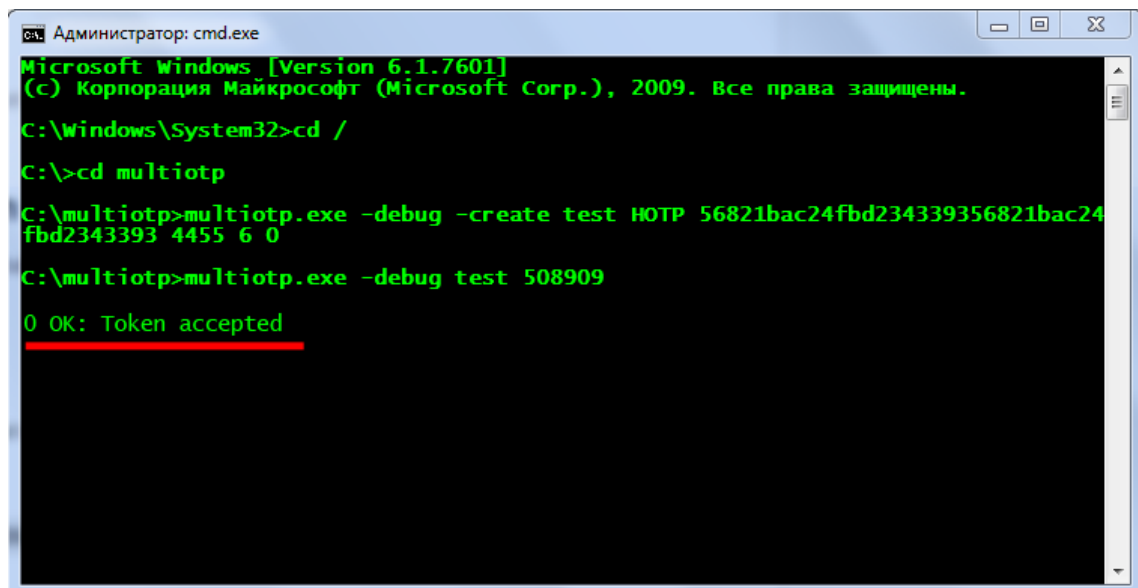
K2BBXLBE7PJDIM4TK2BBXLBE7PJDIM4T

Convert

Figure 20 - Converting hex string to base32 code

source: autor





```
Администратор: cmd.exe
Microsoft Windows [Version 6.1.7601]
(с) Корпорация Майкрософт (Microsoft Corp.), 2009. Все права защищены.
C:\Windows\System32>cd /
C:\>cd multiotp
C:\multiotp>multiotp.exe -debug -create test HOTP 56821bac24fbd234339356821bac24
fbd2343393 4455 6 0
C:\multiotp>multiotp.exe -debug test 508909
0 OK: Token accepted
```

Figure 21 - Token accepted

source: autor

We have been tested two software tokens on Android system: Google Authentication and Android token, which is open source application. Actually now we can continue with registration.

3.1 Registration for web application

- 1) The user enters the login name and password.
- 2) Software or hardware tokens generate random seed number depending on token type (Even token, Time token).
- 3) The user enters the generated seed and registration procedure ends.

This procedure of registration is not very secure. User registration process should take place on the server side with encryption protocol. Indeed, in case of data loss, one time passwords will not be able to protect users' data. This solution is suitable only in our case.

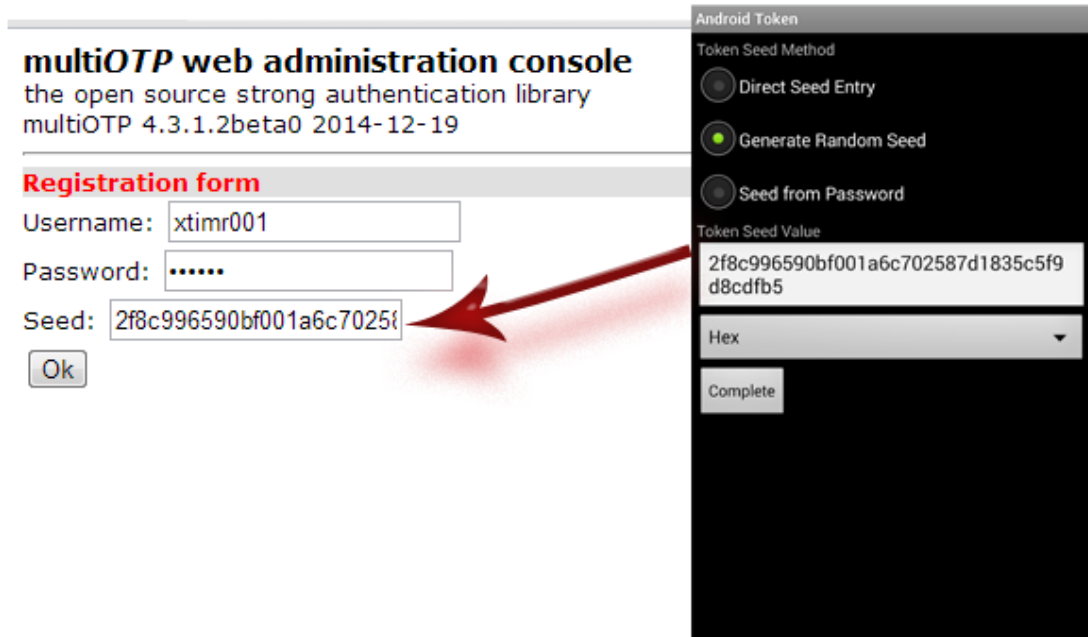


Figure 22 - Example of user registration

source: autor

3.2 Logon

- 1) The user enters a username, password and one-time password, which was generated according to he current seed number.

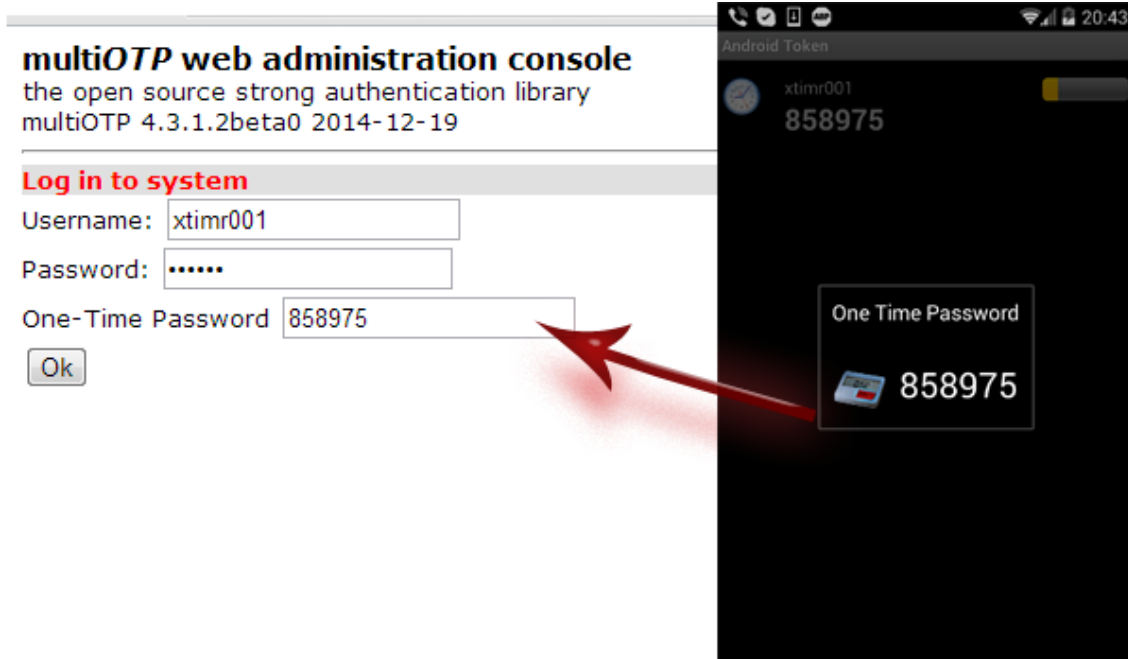


Figure 23 - Example of logon

source: autor

After successful authentication, the user will see the login on panel application.



Figure 24 - Example of successful authentication

source: autor

3.3 Resynchronization existing users

After 6 failed login attempts a user will be blocked. To unlock the user needs to set two one-time passwords. These passwords are based on seed, which can be generated into software token.

multiOTP web administration console

the open source strong authentication library
multiOTP 4.3.1.2beta0 2014-12-19

Logout

- [+] Change admin password
- [+] Import new hardware tokens
- [+] List of hardware token
- [+] Add a new user
- [-] Resync a user**

User to resync:

First OTP:

Second OTP:

Resync now

List of users (1 user)

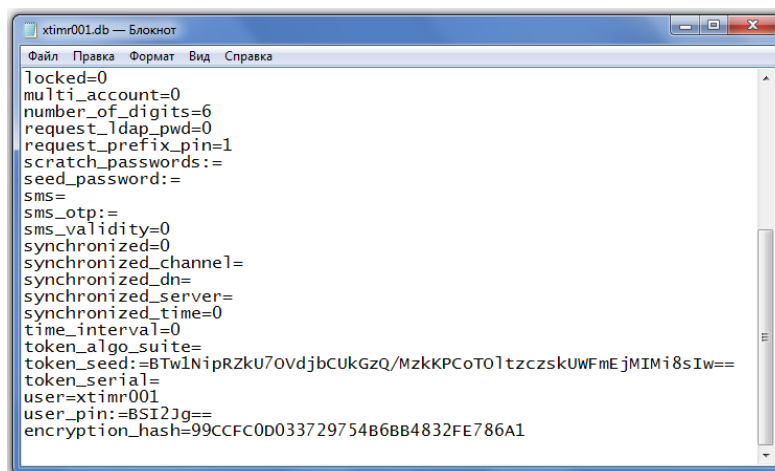
Delete Print Resync xtimr001

Figure 25 - Example of unlocking

source: autor

3.4 Database file

Now we know how the system works. The file *multiotp.class.php*, which is located in utility folder is responsible for all above-mentioned actions. Data of users are stored in the separate file of a database (.db).



```
xtimr001.db — Блокнот
Файл  Правка  Формат  Вид  Справка
Locked=0
multi_account=0
number_of_digits=6
request_ldap_pwd=0
request_prefix_pin=1
scratch_passwords:=
seed_password:=
sms=
sms_otp:=
sms_validity=0
synchronized=0
synchronized_channel=
synchronized_dn=
synchronized_server=
synchronized_time=0
time_interval=0
token_algo_suite=
token_seed=:BTw1NipRZkU70vdjbCukGzQ/MzkkPCoT01tzczsKUwFmEjMIMi8sIw==
token_serial=
user=xtimr001
user_pin=:BSI2Jg==
encryption_hash=99CCFC0D033729754B6BB4832FE786A1
```

Figure 26 - File of xtimr001 user

source: autor

3.5 Types of improving the protection

There are many types of biometric data to protect the information. Each of them has its own security level. There are 2 types of biometric data: psychological and behavioral (fig. 27).

Fingerprint is unique for each person and its hard to forge. To authenticate the user does not need anything extra, except the user's finger, which is very convenient.

Voice is also unique for each user, easier to implement and is a cheaper option than fingerprint scanner.

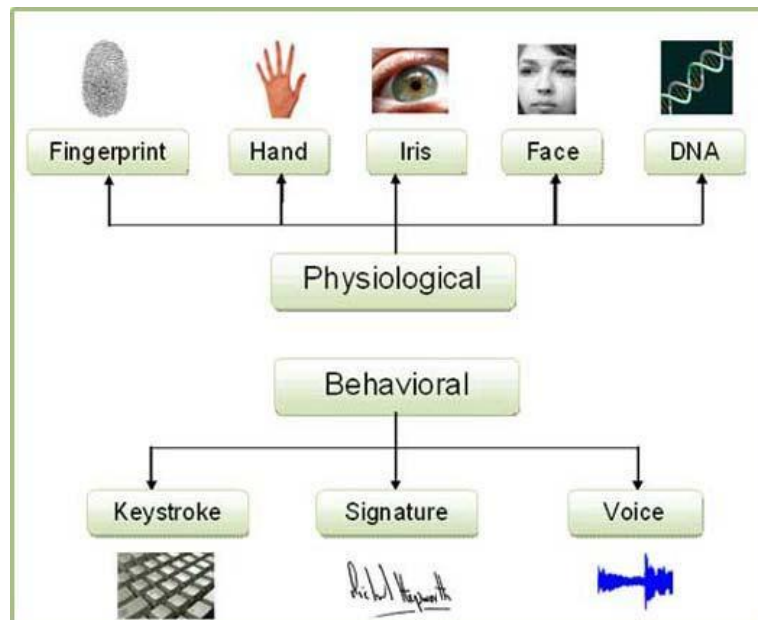


Figure 27 - Types of biometric data

source: itsecurityconcepts.com

According to certain conclusions from theoretical part, we can create the ideal degree of protection in using of internet banking.

Suppose we have some bank account, which use OTP system, we have hardware token that generates a password every 30-60 seconds using a built-in clock, login and our PIN. We enter your login, PIN and our token generates password for us, and we go into the system and can provide some transactions and make manipulation with our account, but your data can be stolen or you can lose your account.

To improve the system - we have to improve the way of protection. Changing of login and PIN does not make sense, because they can be changed only into the system.

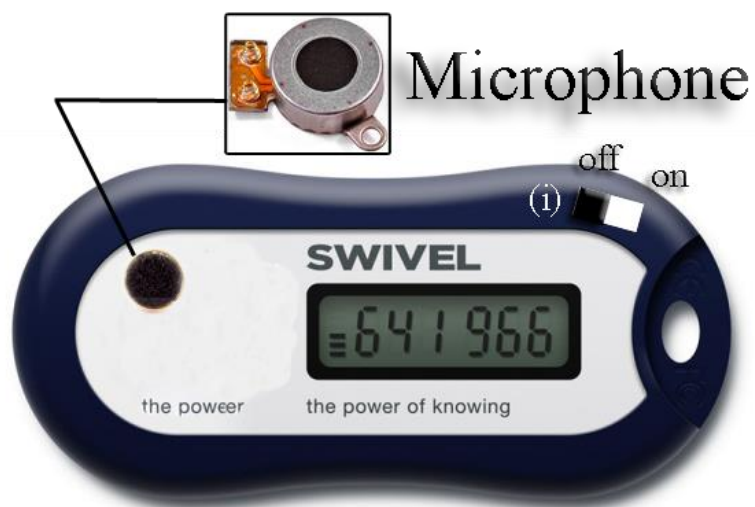
We can improve our hardware token using biometric data of the person. We can integrate things like fingerprint scanners or second variant – microphone (speech recognizer). As you know these two things are unique in each person.



Figure 28 - Prototype of hardware token with biometric data

source: autor

The token does not turn up until the finger touches the scanner and starts comparing of public and private keys. If both key are the same, the display is switched on and the token generates a password that is only valid for 30 or 60 seconds, after that the screen automatically turns off again. To generate a password the user must touch the scanner again and the process repeats. All unsuccessful attempts will be stored on the server.



Another way of integration is the voice recognition via a microphone. The user needs to turn on the microphone using the switcher (i), then the user needs to say passphrase, which is stored on a server on the same principle compare two keys and the display turns on and generates a password.

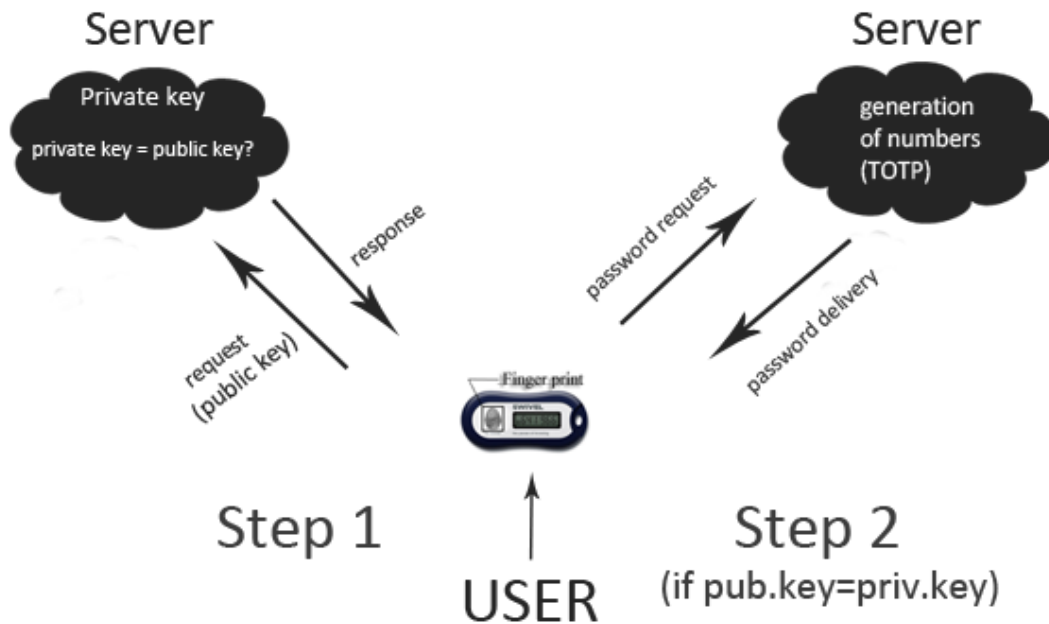


Figure 29 - Scheme of using biometric token for internet banking source: autor

The user enters their biometric data and waits for a response from the server (*fig.29*). If his public key is the same as the key stored on the server (the private key, which user has to create, when he is creating own bank account), the token screen turns on and asks the server to generate the password. After this, server creates a password using the algorithm based on the time, token shows the password and after 30 or 60 seconds off, after that the process starts again.

4. Conclusion

In this Bachelor thesis, I have tried to show the basic principle of use of one-time passwords, their strengths and weaknesses, to find the best distribution OTP method to the user and the best algorithm of generation one-time passwords for different tasks.

In the practical part of the work, I have tried to show the principle of operation of passwords in practice. I used a training tool MultiOTP with PHP library and showed how the user can register in the system using software token. I have shown which file stores all the information of each user. Also tried to make an improvement for Internet banking by integrating into hardware tokens biometric data, which for each person is unique.

One-time passwords are a good solution for authentication with public computers, on which can be installed a program to steal passwords. Authentication using sms messages is very popular among many companies and banks. This is because the mobile phone has almost everyone. Does not require additional resources and costs. However, this type protection is not ideal; in the mobile phone can be mounted virus. For example, Viber or Facebook application automatically opens images from receiver, which can contain malware to steal accounts. Hardware tokens are very good solutions for online banking, however, they can be stolen. A decision, which I cited in the practical part, I think the ideal and very convenient for the protection of any information. I think in the near future, we will completely get rid of hackers attack using our own biometric data.

5. List of figures

Figure 1 - Time synchronized OTP	source: <i>www.slideshare.net</i>	8
Figure 2 - An asymmetric encryption system	source: <i>kryptophone.kryptotel.net</i>	10
Figure 3 - Time Based e-Token	source: <i>www.avfirewalls.com</i>	12
Figure 4 - Time-synchronous scheme	source: <i>cybermashup.com/</i>	12
Figure 5 - algorithm for generating numbers	source: <i>autor</i>	13
Figure 6 - an example of asynchronous authentication	source: <i>autor</i>	14
Figure 7 - S/KEY OTP generator example	source: <i>techpubs.sgi.com</i>	15
Figure 8 - Received message with OTP via SMS	source: <i>www.itrustsecurity.com</i>	17
Figure 9 - Randomly generated grid of pictures	source: <i>confidenttechnologies.com</i>	18
Figure 10 - Credit card token		19
Figure 11 - Waterproof token		19
Figure 12 - Token with biometric data		19
Figure 13 - Usb tokens	source: <i>mcacert.safescrypt.com</i>	20
Figure 14 - Token based on time	source: <i>avfirewalls.com</i>	20
Figure 15 - Token on mobile phone	source: <i>vasco.com</i>	21
Figure 16 - OTP generator installed on PC	source: <i>softpedia.com</i>	22
Figure 17 - Paper token	source: <i>quuxlabs.com</i>	23
Figure 18- A sketch of the KYPS login method	source: <i>en.wikipedia.org</i>	25
Figure 19 - Creating user in system	source: <i>autor</i>	28
Figure 20 - Converting hex string to base32 code	source: <i>autor</i>	29
Figure 21 - Token accepted	source: <i>autor</i>	30
Figure 22 - Example of user registration	source: <i>autor</i>	31
Figure 23 - Example of logon	source: <i>autor</i>	32
Figure 24 - Example of successful authentication	source: <i>autor</i>	32
Figure 25 - Example of unlocking	source: <i>autor</i>	33
Figure 26 - File of xtimr001 user	source: <i>autor</i>	33

6. List of references:

5.1 List of books

ROEBUCK, Kevin. *OpenID: High-impact Strategies: What You Need to Know: Definitions, Adoptions, Impact, Benefits, Maturity, Vendors*. Queensland, Australia: Emereo Publishing, 2012. ISBN 9781743333037.

ARMSTRONG, Damon. *Pro ASP.NET 2.0 Website Programming*. New York: Apress, 2005. ISBN 978-1-4302-0104-5.

CHESWICK, William R a Steven M BELLOVIN. *Firewalls and Internet security: repelling the wily hacker*. Reading, Mass.: Addison-Wesley Publishing Company, c1994, xiv, 306 s. Addison-Wesley professional computing series. ISBN 02-016-3357-4.

CHAKI, Nabendu, Natarajan MEGHANATHAN a Dhinaharan NAGAMALAI. *Computer Networks & Communications (NetCom): Proceedings of the Fourth International Conference on Networks & Communications*. New York: Springer Science & Business Media, 2013. ISBN 1461461545.

BISHOP, Matt. *Computer security: art and science*. Boston: Addison-Wesley, c2003, xli, 1084 s. ISBN 02-014-4099-7.

STEWART, James Michael. *Security fastpass*. San Francisco, Calif.: SYBEX, 2004, xxiii, 197 p. ISBN 07-821-4359-8.

SLONE, John P. *Local area network handbook*. 6th ed. Boca Raton, Fla.: Auerbach, c2000, xiii, 676 p. ISBN 08-493-9838-X.

TODOROV, Dobromir. *Mechanics of User Identification and Authentication: Fundamentals of Identity Management*. Boca Raton, Florida: Auerbach, 2007. ISBN 9781420052206.

BHAIJI, Yusuf. *Etwork Security Technologies and Solutions (CCIE Professional Development Series)*. New York City: Pearson Education, 2008. ISBN 0132796740.

KIM, Tai-Hoon, Hojjat ADELI, John ROBLES a Maricel BALITANAS. *Advanced Communication and Networking: International Conference, ACN 2011, Brno, Czech Republic, August 15-17, 2011, Proceedings*. Verlag Berlin Heidelberg: Springer, 2011. ISBN 978-3-642-23312-8.

LEHTINEN, Rick, Deborah RUSSELL, G GANGEMI a Deborah RUSSELL. *Computer security basics*. 2nd ed. Sebastopol, CA: O'Reilly, c2006, xii, 296 p. ISBN 05-960-0669-1.

5.2 List of electronic sources

MAGALHAES, Ricky. Authentication is changing: passwords have become yesterday's access control. In: *Windowsecurity.com* [online]. 2011 [cit. 2015-02-16]. Dostupné z: http://www.windowsecurity.com/articles-tutorials/authentication_and_encryption/Authentication-changing-passwords-have-become-yesterdays-access-control.html

ROUIT, Olivier. OTP (One Time Password) Demystified. In: *CodeProject - For those who code* [online]. 2013 [cit. 2015-02-16]. Dostupné z: <http://www.codeproject.com/Articles/592275/OTP-One-Time-Password-Demystified>

MAJOR, Peter. One-Time Passwords – HOTP and TOTP. In: *ForgeRock Community Blogs* [online]. 2014 [cit. 2015-02-16]. Dostupné z: <http://blogs.forgerock.org/petermajor/2014/02/one-time-passwords-hotp-and-totp/>

SULLIVAN, Tiffany. SMS is the Weakest Channel to Deliver OTP. In: *Strong Authentication, Two-Factor Authentication* [online]. 2013 [cit. 2015-02-16]. Dostupné z: <https://www.wypass.com/sms-is-the-weakest-channel-to-deliver-otp/>

PRINCE, Brian. Facebook Adds One-Time Password Security Feature to Protect Accounts - See more at: <http://www.eweek.com/c/a/Security/Facebook-Adds-OneTime-Password-Security-Feature-to-Protect-Accounts-890325#sthash.C4qyI4qs.dpuf>. In: *Technology News, Tech Product Reviews, Research and Enterprise Analysis - eWeek.com* [online]. 2010 [cit. 2015-02-16]. Dostupné z: <http://www.eweek.com/c/a/Security/Facebook-Adds-OneTime-Password-Security-Feature-to-Protect-Accounts-890325>

7. List of tables

Table 1 - Comparison of RSA hardware tokens source: emc.com24

8. Annex: Part of the code of MultiOTP

Creating a new user:

```
require_once('multiotp.class.php'); // Insertion of PHP MultiOTP library
$multiotp = new Multiotp('MyPersonalEncryptionKey'); // Creation of a new key (after
creating the class without argument is DEPRECATED)
$multiotp->EnableVerboseLog(); // Logging of actions (could be helpful at the
beginning)
$multiotp->SetUser('username'); // Creation of a new username
$multiotp->SetUserPrefixPin(0); // If we don't want the prefix PIN feature leave this
line empty
$multiotp->SetUserAlgorithm('TOTP'); // Type of algorithm (TOTP or HOTP)
$multiotp->SetUserTokenSeed('seed');
$multiotp->SetUserPin('pin'); // Insertion of PIN or password
```

```

$multiotp->SetUserTokenNumberOfDigits(6); // Quantity of numbers (usually 6)
$multiotp->SetUserTokenTimeInterval(30); // On the expiration of time of number of a
token will change
$multiotp->WriteUserData(); // Adding a user to the database

```

Verification a token:

```

require_once('multiotp.class.php');
$multiotp = new Multiotp('MyPersonalEncryptionKey');
$multiotp->EnableVerboseLog();
$multiotp->SetAttributesToEncrypt('*user_pin*token_seed*token_serial*seed_passwor
d*');
$multiotp->SetUser('username');
if (0 == $multiotp->CheckToken('token'))
{
// Authentication accepted
}
else
{
// Authentication rejected
}

```

Resynchronization a user:

```

require_once('multiotp.class.php');
$multiotp = new Multiotp('MyPersonalEncryptionKey');
$multiotp->EnableVerboseLog();
$multiotp->SetUser('username');
if (0 == $multiotp->CheckToken('token1','token2')) // it must two consecutive tokens
{
// Synchronization successful (getting number 14)
}
else
{ // Synchronization failed (getting number 27) }

```