

A PingPong One-Time-Password system in Java application

Bayalagmaa Davaanaym¹, Young Sil Lee¹, HoonJaeLee¹, SangGon Lee¹
and HyoTeak Lim¹

¹Department of Ubiquitous and IT
Graduate School of Design and IT, Dongseo University
Busan, 617-716, South Korea Tel: +82-51-320-1730,
E-mail: bayalag2007@yahoo.com, [hjlee](mailto:hjlee@dongseo.ac.kr), [nok60](mailto:nok60@dongseo.ac.kr), [htlim](mailto:htlim@dongseo.ac.kr)

Abstract. The several techniques using technology based on biometrics, passwords, certificates, and smart cards can be used for user authentication in the accessible network system. One of the most popular areas in OTP authentication protocol can be used for authenticating a user by a server. It increases security by using a new password for each authentication while the previous password scheme iteratively uses a same password. In this paper we propose a secure user authentication protocol with one time password based on PingPong128 stream cipher .

Keywords: One time password, Stream cipher, PingPong128

1 Introduction

More and more services are becoming available on the Internet, and many of these services require authentication. The most widespread solution for electronic authentication today is the use of a username and password. As the number of services grows so does the number of username and password pairs that the user needs to remember. Already many people are experiencing that it is impossible to remember all the combinations and therefore they use the same pairs for all their services and choose passwords that are easy to remember. This strongly reduces the security of an already weak authentication mechanism.

Several more secure solutions for electronic authentication exist, such as One-Time-Password (OTP) or Smart Card PKI solutions. They solve the security problem with passwords, but they most often increase the burden for the user. Extra hardware and software is required both for the user and service provider and they are often specific for each service so the user needs to deal with many different devices and procedures.

One-time password when the user demand an authentication, password which always creates with the method which it uses is having each other different values and once the password which is used to prevent reuse. The attacker was a eavesdropping on the network or the user password is forgotten the password, even if safety can be

guaranteed. Such one time passwords were normally used for Internet banking, but recently are also used when purchasing online games, music and videos. The demand for one time passwords is increasing and they are being used in more diverse fields. This study presented a password key creating method using Ping Pong128 stream cipher that create a strong new OTP key generator on web/mobile authentication.

In this paper we propose a web/mobile authentication system with one time password that is both secure and highly usable, based on multifactor authentication approach. It uses a novel approach to create an authentication system based on IC's (Identification code) and message context. IC's are user specific unique transmission identification codes which are issued by banks, companies or financial institutions to the user. Ayu Tiwari et al[4] have also proposed the use of SMS with used Transaction Identification codes in their protocol but its code is similar to One Time Password (OTP) and code is used only once that would be used to keep IC's as secret codes on devices or mobile phones. In our approach authentication server generate only one time a secret code on devices and next round it will be change secret code value.

Chapter 2 of this study explains the OTP and PingPong128 stream cipher related study, and chapter 3 presents a password key creation method through extraction of PingPong-128 stream cipher. PingPong-128 is strong stream cipher function, it can create temporary one time password keys. Chapter 4 carries out a security analysis the presented one time password key algorithm, and lastly, draws a conclusion.

2 Related works

Normally when a user requests authentication, even after first contact, certain important services confirm passwords again. However, as explained above, the existing password system has many weaknesses, and a solution for this is one time password system. One Time Password (OTP) is a password system where passwords can only be used once, and the user has to be authenticated with a new password key each time. It is a password key creation method that makes it extremely difficult to predict the next password key based on the current password. A new password key is created in its own device constantly after a set period of time and the user has to enter a new password every time he or she uses the system, so it prevents exposure of the user's password due to hacking or the user's mistake.

OTP has much stronger security because the user has to enter a newly created password key even if his or password is exposed. The OTP is standardized by the IETF, and standardized again by verification related companies, and the RSA[11] and OATH[12] are carrying out the most competitive standardizations.

Within the combined web/mobile authentication, various studies have been made and different solutions have been proposed.

In one of the solutions tailored for the electronic payments [9], the communication protocol relies on SMS messages. These messages have various drawbacks: they can experience strong delays since they are not real-time; they have a limit of 160 characters and there is a privacy issue because the seller needs to know the phone number of the buyer. Another solution in [10], uses the authentication algorithm of the cards Subscriber Identity Module (SIM). Every SIM contains a secret key, known only by the authentication center which has released the SIM. Without getting into details,

one can meet security problems and various other issues related to the contractual agreements between telephone carriers and sellers.

To resolve the weaknesses in the previous systems, an architecture presented multifactor authentication approach [4]. The use of more than one authentication method is referred to as multi-factor or strong authentication, the most common of which is two-factor authentication. A familiar example of two-factor authentication is an ATM card, where, in order to access your account, you insert the card (something you have) into the ATM machine and enter your PIN (something you know). Implementing multi-factor authentication has been growing in popularity as organizations look to increase security and meet the demands of industry and government regulations that require protection of sensitive consumer and employee information. Most organizations will typically already have a user name and password system in place for network authorization and access. Deploying a USB token or smart card solution is quickly becoming the method of choice for achieving increased security. Many factors contribute to the popularity of multi-factor authentication, including cost, convenience, and user acceptance.

2.1 Secure Authentication protocol

In this electronic age, where identity and data theft are on the verge of becoming commonplace, it is vital that a person's digital identity be trusted at all times. To achieve this, a level of challenge can be invoked in the authentication process, forcing the user to prove their authenticity beyond a simple password. The more factors used to determine a person's identity, the greater the trust of authenticity.

In terms of security, authentication is distinctly separate from authorization, which provides access to specific applications and data based on the user's identity. Authentication ensures that the user is who she claims to be, while authorization defines her role once they are granted access.

Single-factor authentication is inadequate for high-risk transactions involving access to customer information or the movement of funds to other parties. Multifactor Authentication techniques can be used to provide secure web transactions using mobile phones and other devices.

Various methods have been proposed to provide multifactor authentication, like biometrics, extra hardware and software etc. However, the most practicable ones are those based on two separate communication channels. This technique is used in the present work. In the present work, we propose a multifactor authentication technique based on IC's files.

OTP Authentication: Another method to validate user account is OTP confirmation. The financial institution stores user cell phone number to provide multi-factor authentication [4]. We assume that users carry their mobile phone with them regularly and therefore can access web authentication server that get one time passwords by mobile internet. As a result, only valid users will get an OTP from the authentication server. After getting the OTP, a user can access the web.

When the authentication server receives right OTP it knows that the user is valid and that the user has.

IC Authentication: IC authentication is the technique which is used to verify both the user and the ongoing transmission. A IC code certifies that the current transmission has been initiated by the right person and that it is a valid user who is trying to access his/her account. IC codes are:

- Issued by the authentication server to its customers.
- An 8 bit or 16 bit Pseudo randomly generated code which is assigned to the customers.
- May be a complicated digit sequence or combination of numeric characters.

In our protocol the user will get a initiate IC code from the server according to their requirement. IC code is encrypted and stored on user's phone application. The server will keep records of issued IC codes to its customers and match the same code during the online web transaction. A IC code value is changed after each transaction on server and phone application. Server can also decide to designate validity time period of ICs according to its standard organizational issuing policies, providing an extra security feature.

So, Multifactor Authentication is used to verify the user and the transaction by using following steps:

1. **Web-Based Basic Authentication:** Firstly, the user has to prove his identity to the Web server using a Web-based username/password based protocol for basic authentication.
2. **OTP Authentication:** After authenticating the web user using username/password, the user get one time password from Web server. Now user will insert one time password code to the web Authentication server.
3. **IC Authentication:** After authenticating the web user using username/password, the Web server demands a IC code from the web user. Now user will decrypt and insert one time IC code to uniquely identify his/her transaction and prove his/her identity to the web Authentication server. Authentication server to confirm his/her transmission.

2.2 Description of Ping Pong-128 Stream cipher

The elements of one time password system are a token included in a security/password algorithm or one time password key creating device, a authentication server and a authentication client. Since the one time password mechanism is also a program, it is programmed to be random, but the randomness breaks after a certain period of time and passwords become predictable so one time password mechanisms have the disadvantage of having to exchange OTP modules after a certain period of time. To handle key management, the idea is to rely on a one-way hash function such as MD4, MD5 and SHA. Some weaknesses have recently been discovered in the MD5 and SHA-1 algorithms.

The hashes are designed so it is very difficult to find two messages that produce the same hash, this is called "collision resistance". Because MD5 is 128-bit, by random chance you will find a collision by producing 2^{64} hashes. The weakness in MD5 is that a way has been found to produce such collisions with only 2^{42} hashes. This makes producing collisions practical and I have seen an example of 100 different collisions. In order to overcome such weaknesses, this study presents a method of creating one time password keys in OTP Clients using Ping Pong-128 stream cipher.

Here we explain the Ping Pong128 stream cipher [1] and some analysis of generator. Stream ciphers are developed as approximation to behavior one time code. The one-time password uses a long string of key stream which consists of bits that are chosen completely at random. This key stream is combined with the plaintext on a 'bit by bit' basis. The key stream is the same length as the message and can be used only once clearly a vast amount of key stream might be required.

PingPong-128 is a specific cipher from the Ping Pong family of stream ciphers. It has two mutually clocking LFSRs and a single memory bit

Of the two styles of LFSR, the usual style is called a Fibonacci LFSR. To shift a Fibonacci LFSR, simply copy each bit to its neighbor on the right. The original rightmost bit is considered the output. The bit that is shifted in at the left is the parity of some specific subset of the bits of the register. The most important properties of an LFSR are that it has a low gate delay, and more importantly if the taps are chosen properly, repeated shifting will cycle through every possible non-zero value of the register. The other style of LFSR is called a Galois LFSR, which has the same properties as the Fibonacci LFSR, but is shifted differently. To shift a Galois LFSR, copy each bit to its neighbor on the right, except for the taps, for which the right most bit of the register is XOR'd in before the copy is done. The bit that is shifted in at the left is the original rightmost bit, which is also considered the output. Implementation issues both LFSRs in PingPong-128 use the Galois LFSRs rather than the Fibonacci LFSRs.

The Ping Pong LFSRs are of lengths 127 bits and 129 bits. Together with the memory bit they give PingPong-128 an internal state of 257 bits. PingPong-128 takes a 128-bit key and a 128-bit initialization vector to fill the internal state. Keystream generation The PingPong generator produces the output keystream by combining the LFSR sequences and the memory sequence

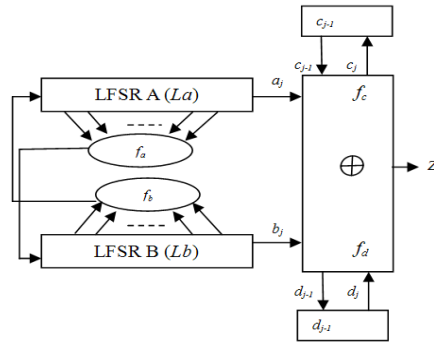


Fig 1. PingPong Family generator

PingPong-128 have a two primitive polynomials $P_a(x)$ and $P_b(x)$ are following here :

$$P_a(x) = x^{127} \otimes x^{109} \otimes x^{91} \otimes x^{84} \otimes x^{73} \otimes x^{67} \otimes x^{66} \otimes x^{63} \otimes x^{56} \otimes x^{55} \otimes x^{48} \otimes x^{45} \otimes x^{42} \otimes x^{41} \otimes x^{37} \otimes x^{34} \otimes x^{30} \otimes x^{27} \otimes x^{23} \otimes x^{21} \otimes x^{20} \otimes x^{19} \otimes x^{16} \otimes x^{13} \otimes x^{12} \otimes x^7 \otimes x^6 \otimes x^2 \otimes x \otimes 1$$

(1)

$$P_b(x) = x^{129} \otimes x^{125} \otimes x^{121} \otimes x^{117} \otimes x^{113} \otimes x^{109} \otimes x^{105} \otimes x^{101} \otimes x^{97} \otimes x^{93} \otimes x^{89} \otimes x^{85} \otimes x^{81} \otimes x^{77} \otimes x^{73} \otimes x^{69} \otimes x^{65} \otimes x^{61} \otimes x^{57} \otimes x^{53} \otimes x^{49} \otimes x^{45} \otimes x^{41} \otimes x^{37} \otimes x^{33} \otimes x^{29} \otimes x^{25} \otimes x^{21} \otimes x^{17} \otimes x^{13} \otimes x^9 \otimes x^5 \otimes 1 \quad (2)$$

PingPong-128 takes a 128-bit key and a 128-bit initialization vector to fill the internal state [2]. The PingPong128 OTP generator produces the output key stream by combining the LFSR sequences and the memory sequence. PingPong-128 has two clock-control functions, $f_a(L_a)$ and $f_b(L_b)$, and the output keystream bit Z and memory bit c at time j are defined to be identical to the summation generator:

$$f_a(L_a) = 2L_{a42}(t) + L_{a85}(t) + 1 \quad (3)$$

$$f_b(L_b) = 2L_{b43}(t) + L_{b86}(t) + 1 \quad (4)$$

During this phase the whole keystream is generated constantly depending on the key and its past. The process is realized applying two similar functions named a_j and b_j , and defined as follows:

$$c_j = f_c(a_j, b_j, c_{j-1}) = a_j \otimes b_j \otimes (a_j \otimes b_j) c_{j-1} \quad (5)$$

$$d_j = f_d(a_j, b_j, d_{j-1}) = b_j (a_j \otimes b_j) d_{j-1} \quad (6)$$

$$Z_j = f_z(a_j, b_j, c_{j-1}, d_{j-1}) = a_j \otimes b_j \otimes c_{j-1} \otimes d_{j-1} \quad (7)$$

PingPong128 has a pair of LFSRs of different lengths where used a number of different feedback polynomials and took clocking taps from various stages of the registers. As a result, It is resistant against attacks based on basic key stream properties such as linear complexity and period. It defeats time-memory tradeoff and known attacks against the summation generator and other clock controlled key stream generators.

3 Password keys using stream cipher

The password key creation process starts with the user send a request to server. Then next process shows the randomly prime number selecting and reply to client. The process of creating a combination of permutation using the selected prime number by order, and creation of an OTP keys with Ping Pong128 hash function. OTP password generation begins with a secret key. This secret key can either be provided by the user, or can be generated by a computer.

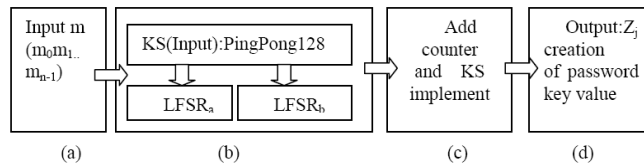


Fig.2 Password Key creation process design

The password key creation process starts with the user insert his or her challenge to server and after the verified user information, OTP key generator gets plain password from server, as shown in (a) of Figure 2. This challenge is not used for one time password authentication. The range of this study starts after the authentication process is over, and is not related to authentication. However, another advantage is that it can be used together with another web-based authentication system. Existing authentication systems can be added to this study to create a strong new OTP key generator.

Key stream algorithm, key loadings and key renewal parts many do the resources and operation in necessity and KS() function, the encryption/decryption will be for the purpose of generating one time password. The key bit codes for generation are as follows: Key stream algorithm KS()

```

{
    fa = 2*LFSR1[42]+LFSR1[85]+1;
    fb = 2*LFSR2[43]+LFSR2[86]+1;
    for(iiii=0; iiii<fb; iiii++)
    {
        LFSR127();
    }
    for(iiii=0; iiii<fa; iiii++)
    {
        LFSR129();
    }
    zj = (byte)(LFSR1[126]^LFSR2[128]^cj^dj);
    cj = LFSR1[126]*LFSR2[128]^(LFSR1[126]^LFSR2[128])*cj1;
    dj = LFSR2[128]^(LFSR1[126]^LFSR2[128])*dj1;
    return (byte) zj;
}

```

As shown in (b) of figure 2, the PingPong128 key loading and key updates based on the two LFSR, will fill input value. (c) of figure 2 shows the process of move to next bit round in KS() function as follows:

$$y_n = m_n \oplus z_n \text{ for } 0 \leq i \leq n-1 \text{ where } \oplus \text{ denotes bitwise exclusive-or.} \quad (8)$$

d) of figure 2 input value of the user plain password, and will be used, such as the structure of Figure 2, the operations to generate the Z_j value is the OTP key value of output.

$$\begin{aligned}
 &: \\
 &y(1) = m \oplus z \\
 &y(2) = (m^1 \oplus z^1 || y(1)) \\
 &y(3) = (m^2 \oplus z^2 || y(1) || y(2)) \quad \dots \\
 &y(n) = (m^{n-1} \oplus z^{n-1} || (y_{n-2} .. (y_{n-(n-1)}))) \\
 &\text{Keystream} = y(1) || y(2) .. y(n)
 \end{aligned} \quad (9)$$

PingPong-128 hash function is applied n times to secret key, thereby producing a hash chain of n one-time passwords values. The initial secret key is discarded.

The user is provided with the only one password, printed out in reverse order. Only the one password, at the user's list, is stored on the server. When the server receives the y_n one time password from the applet it checks that it matches the one time password it created itself. If the values match the user is successfully authenticated. If the solution is part of a single sign-on environment an assertion is created that can be reused for several authentication procedures. The next section gives the complete protocol proposed based on the above technique

4 Authentication protocols with OTP

The data flow and architecture, based on Multifactor Authentication techniques, is described in this section.

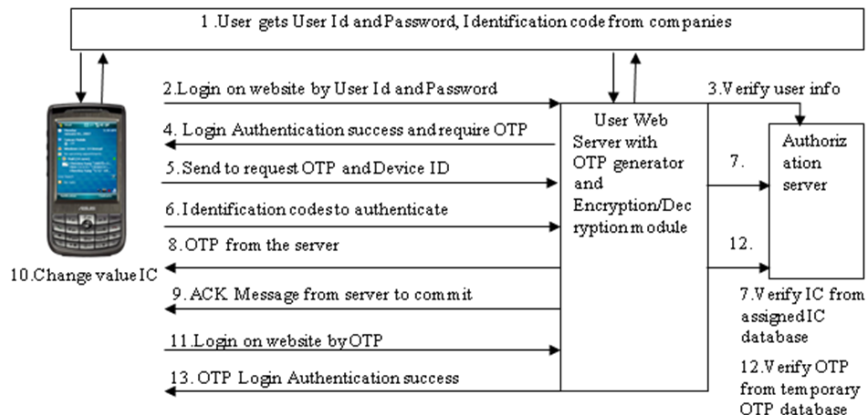


Fig 3. Multifactor secure Web authentication protocol using mobile devices

Below we describe each step of the above protocol.

- 1.) User gets username/password and Identification codes (ICs) from the Companies. Each user has only one username/password to their account, but IC code is unique for each online transmission. So users will automatically gets list of IC codes from the company authority according to their requirement.
- 2.) A Web-based username/password basic authentication is used to identify the user to the Web server.
- 3.) The username and password will be verified by the Authentication Server. After user recognition the user will get option screen to proceed further
- 4.) The user will get a notification of a successful logging with require OTP message.
- 5.) The user will send to request.
- 6.) The user mobile will automatically insert a IC code. Note that IC is stored, with password protection, with a local encryption on the cell phone. The user will first time insert the IC form company to next time it will be automatically give unique authentication to each web transaction. All details

of the operation, with attached IC, will be further encrypted by AES encryption technique and submitted to the web server. The web server would pass it on to the authentication server where it would be decrypted and matched with the list of ICs that have been issued to the user.

- 7.) The authorization server decrypts the received message and extracts the IC. It then verifies the IC received from the user by comparing it with the stored list of ICs in the user account information at server database. If both ICs match then it cancels the used IC from its database and goes to the next step. If no IC matched with those in database then the authentication server will deny the user transaction and display appropriate error message to the user.
- 8.) If IC authentication will success the server will generate PingPong based One time password using secure password from user database and send to user mobile phone
- 9.) The server will notify the user by a Message to acknowledge the successful completion of operation. Change IC values in the database. Change IC values in the mobile .

4.1 Implementation

J2ME is the preferred development platform due to the following reasons: the portability of Java code, the ability to reduce the network traffic by making use of the processing power of the Java phone to process data locally, and the ability to establish a differential security policy on the client that will perform the encryption operations according to the content and sensitivity of network data rather than encrypting everything.

Also, J2ME mobile information device applications (MIDlets) can operate over and make use of, the WAP stack to perform HTTP network interaction, without requiring TCP/IP. A side-effect of devising a general application-layer security solution using J2ME is that it provides a feasible solution to the traditional security gap in the WAP gateway [17].

This security gap is due to the security protocol conversion mechanism taking place in the WAP gateway between the secure sockets layer (SSL) encryption and the WAP wireless transport layer security (WTLS) encryption protocols. This protocol conversion leaves data in an unencrypted form at the gateway during the protocol switching process, which increases the risks to the confidentiality of data in the gateway [18]. HTTP is a stateless protocol. Each HTTP request is independent of the other requests and the protocol specification does not devise any mechanism that can group a series of requests as belonging to one user or another. Over time, several strategies have evolved to address session tracking; the most practical of these are the use of cookies and URL rewriting. The Java Servlet API provides the HttpSession object, which represents each user's interaction with the web server. The Servlet API specification requires that web “containers” implement session tracking using the cookie mechanism. The cookie interchange mechanism is similar to that taking place between a web browser and a web server with one difference – in our proposed protocol, the client MIDlet program has to explicitly send the session cookie back to the server for every request.

Our simulation of client applications has used The Sun J2ME Wireless Toolkit consisting of build tools, utilities and a device emulator. The authentication server is based on J2EE technology with web server Glassfish and database Mysql. Module Ping Pong One time password codes can be generated with secure password from user database we explain previous section. And Module IC codes are pseudo random codes and can be generated with pseudo random number generation algorithm .

The authorized person is responsible for the distribution of IC's to the user cell phone via simple internet browser or a Bluetooth device and distribution process includes the encryption of IC's for security reasons.

At server side, we have assumed IC's and secure passwords are stored in database and there is strong security of Database management system and operating system with secure firewalls to protect server side data. The implementation of this protocol will not increase expenses of users significantly. This protocol can be easily implemented and executed on the current expenses charged by server from the users.

5 Security analysis

Since a comprehensive security analysis has been provided on authentication with web/mobile, and PingPong OTP key and key stream in this research.

5.1 Analysis of key and keystream

The key of the stream cipher is obtained using the function PingPong128. For PingPong-128, both key K and IV have a length of 128 bits, and together they fill 257 bits of internal state. The initialization process can also be used for rekeying. The process to generate the initial state for the key stream generator uses the generator itself twice.

The starting state of LFSR_a is obtained simply by XOR -ing the two 128-bit binary strings of the key, K, and IV. The first iteration cipher is run to produce an output string of length 257 bits. For the second iteration of the cipher, the first 128 bits of this output string are used to form the initial state of LFSR_a, and the remaining 129 bits are used to form the initial state of LFSR_b.

The cipher is run a second time to produce an output string of length 257 bits. The output from this second application is used to form the initial state of the key stream generator when we begin key stream production.

By employing the Ping Pong algorithm itself, we take advantage of both the known security properties of the algorithm and where K₁ and K₂ derive using the compression function γ of the hash function h and they cannot be distinguished by the attacker from truly random keys. PingPong128 has a pair of LFSRs of different lengths that pair used a number of different feedback polynomials and took clocking taps from various stages of the registers. It was observed that the choice of feedback polynomials and clocking tap position did not influence the key stream properties.

The design strength of PingPong-128 is 2^{128} . It is therefore resistant against attacks based on basic key stream properties such as linear complexity and period.

But, a weak pseudo-randomness of γ is required since the attacker that is trying to

find out the dependencies of K_1 and K_2 cannot see directly the output of the pseudorandom function on any input. In summary, as claimed by Hoon Jae Lee, Kevin Chen in [3]: "It defeats known attacks against the summation generator and other clock controlled key stream generators".

5.2 Analysis of authentication protocol

Even though not strictly necessary, a specific security mechanism has been implemented on the connection through the encryption of any exchanged data by AES algorithm. AES is also used in the challenge/response procedure and in the dispatch of the Ping Pong OTP to the Mobile Device. As illustrated, AES is used with two elements: the secret key and the random data generated by the Authentication Server. The usage of more communication channels .

In a real scenario, a phishing attack would fail because, even with the unauthorized usage of a user's basic authentication credentials, it would be impossible to complete all the steps involved due to the lack of the needed data, such as the secret password key and user identifier that are stored in the Authentication Server. The proposed system could present some critical points related to the Support Server and Authentication Server security.

6 Conclusion

Internet security is a recent important point of interest with the wide use of the Internet. Security systems against improper actions only perform well when all the parts consisted create perfect harmony. An important aspect of this work is to consider the Ping Pong one time password with authentication system.

The hash function of one time password system can be seen as a module that can be replaced in case serious weaknesses are found in the hash function or when new more secure or efficient hash function are designed. This study presented an OTP model with a password key creating method using Ping Pong128 stream cipher, and an OTP model that can adopt a password key creating algorithm that improve multifactor authentication protocol.

References

1. Hoonjae Lee, Kevin Chen "PingPong-128, A New Stream Cipher for Ubiquitous Application" IEEE CS (ICCIT2007), ISBN: 0-7695-3038-9, pp.1893-1899, Kyungju, Nov. 21-23, 2007.
2. HoonJae Lee, Il Seok Ko (Franz Ko) "An Intelligent Security Agent for Reliable Cipher System using PingPong," Cybernetics and Systems Journal, Vol. 39, No. 7, pp.705-718, Oct.-Nov., 2008. ISSN 0196-9722

3. Yoon Eun-Jun, Yoo Kee-Young "One-Time Password Authentication Scheme Using Smart Cards Providing User "International Conference on Computational Science and its Applications, Glasgow , ROYAUME-UNI 2006 , vol. 3984, pp. 303-311
4. Ayu Tiwari, Sudip Sanyal, "A Multifactor security protocol for wireless payment-secure web authentication using mobile" IADIS2007, ISBN: 978-972-8924-30-0,pp.160–167, Feb. 18-20, 2007
5. N. Haller and R. Atkinson "On Internet authentication," Internet Request For Comments 1704, Oct. 1994.
6. T. Tsuji and A. Shimizu, "An one-time password authentication protocol OSPA," IEICE Trans. Commun., Vol.E86-B, No.7, pp. 2182-2185, 2003
7. T. Tsuji and A. Shimizu, "One-time password authentication protocol against theft attacks," IEICE Trans. Commun., Vol.E87-B, No.3, pp. 523-529
8. C.M.Chen,W.C.Ku "Stolen verifier attack on Two New Strong Password authentication protocol" IEICE Trans. Commun., Vol.E85-B, pp. 2519-2521,2002
9. J. Preneel B., and Vandewalle J., "Combining world wide web and wireless security", Proceedings of IFIP I-NetSec, Leuven (Belgium), 2001.
10. X., Huang G., Min Z., Wenyin L., Xiaoyue L., "Detection of Phishing Webpages based on Visual Similarity", International World Wide Web Conference, Chiba (Japan), 2005.
11. RSA, <http://www.rsa.com>
12. OATH, <http://www.openauthentication.org>
13. Hoonjae Lee, Sangjae Moon, "On An Improved Summation Generator with 2-Bit Memory," Signal Processing, 80(1), pp. 211217, Jan. 2000
14. B. Schneier, Applied Cryptography, 2nd Edition, Wiley, New York, 1996.
15. T. Tsuji and A. Shimizu, "Cryptanalysis on one time password authentication schemes using counter value", *IEICE Transactions on Communications*, E87-B(6):1756–1759, June 2004.
16. G.Tsudik. "Message authentication with one-way hash functions". ACM Computer Communications Review, Vol.22, No. 5, 1992, pp. 29-38.
17. Soriano M. and Ponce D., (2002), A Security and Usability Proposal for Mobile Electronic Commerce, IEEE Communication Magazines, Vol. 40, pp. 62- 67.
18. Itani Wassim and Kayssi Ayman I., (2004), J2ME End-to-End Security for M-Commerce, Journal of Network and Computer Applications, Vol. 27, pp. 13-32.