Ciência e Natura
ISSN: 0100-8307
cienciaenaturarevista@gmail.com
Universidade Federal de Santa Maria
Brasil

Mashhadi, Najmeh
Authentication in mobile cloud computing by combining the tow factor Authentication and one time password token
Ciência e Natura, vol. 37, núm. 6-2, 2015, pp. 220-229
Universidade Federal de Santa Maria
Santa Maria, Brasil

Ciência e Natura, v. 37 Part 2 2015, p. 220–229

ISSN impressa: 0100-8307 ISSN on-line: 2179-460X

ciência e natura

Authentication in mobile cloud computing by combining the tow factor Authentication and one time password token

Najmeh Mashhadi

Department of Computer Engineering, Islamic Azad University, Science and Research Branch, Tehran, Iran.

Abstract

The Cloud has become a popular business transaction platform nowadays. Unfortunately, this powerful and pervasive network somehow is overshadowed by the growing security threat emerging from the various attacks Authentication is One of the major security issues in mobile cloud computing. Combinig the Two-factor Authentication (2FA) technology with One-time Password (OTP), has emerged as a popular protection system. The 2FA system employs two user specific factors for authentication. It can significantly enhance the network security. We used a dynamic one time password as a second factor. These otp codes provide strong security and resist MITM-seed tracing and shoulder surfing attacks.

Keywords: *Mobile cloud computing, Authentication, 2FA*

Recebido: dia/mês/ano Aceito: dia/mês/ano

1 Introduction

Cloud computing is a model that provide on time and on-demand computational services with suitable price for users. As an evolution of service-oriented architecture and virtualization, the cloud computing potentially provides infinite computational capabilities for its customers, with a method of pay-as-you-consume, which is far more different from the traditional way of network computing. This is an exciting combination for many researchers. Some of them even predict that it might inspire our research in mobile computing over the next decade and beyond.

1.1 Mobile cloud computing

mobile and handheld devices are constrained due to resource limitations primarily caused by limited battery life requiring recharging, constrained size of memory or limited power of the processor especially during roaming and challenge of being seamlessly connected throughout mobility or even limited size of physical persistent storage. Execution of high computational tasks in a mobile device may also drain the battery power very quickly. To address these limitations of mobile devices, cloud computing can be an obvious choice, This means that mobile users offload processing intensive and storage demanding portion of mobile application from resource constraint mobile device to resource enriched cloud. The offloading of the processing intensive and storage demanding portion(s) of mobile application enhances the capabilities of mobile devices in term of processing, storage, and battery.

Cloud computing with its pay-per-use mode alleviates the in-house computing cost and thereby stands as an obvious choice for global enterprises targeting cost optimization yet having flexible, secured and efficient use of IT resources. The main objectives of the mobile cloud computing are to increase the processing/storage capabilities of the mobile device and reduce the energy consumption while executing the computationally intensive jobs [1].

Cloud Computing has some security issues such as virtualization technology security, massive distributed processing technology, service availability, massive traffic handling, application security, access control, and authentication and password. User authentication among them requires a high-quranteed security. Hence, this paper would like to discuss technologies of access contol and user authentication briefly and look at the problems inside. Although many security algorithm represented in mobile cloud computing, but all of them can not implement directly in mobile device, all of them dont considering the limitation of the mobile device. then we need a lightweight scheme that provide security with minimal power consumption and low communication overhead. In this article we focus on secure authentication system in Mobile cloud computing, That consider the Mobile device limitations and Resist Against various attacks.

1.2 Proposed solution

In this article, we use two-factor authentication protocol (2FA)[2] to secure access to remote cloud. 2FA's operation is as follows. A user sends her UserID and password to the network authentication sever for authentication. The server verifies the user by her UserID and password in server's database. If it is matched, the server then asks the user to send the correct prior registered second factor for final authentication. Once being fully authenticated, the user is allowed to access the network. We used a dynamic password as a second factor. This dynamic and one time password decrypted by using Software OTP token functions under the Oauth standard[3]. Oauth's mission is to develop an open reference architecture by leveraging the existing open security standards for the universal adoption of the strong authentication system. Oauth addresses the standardization, compliance and interoperability issues associated with OTP algorithm. But it does not address the issues on implementation method, cost, system deployment plus maintenance, secret key protection and user experiences of the OTP token. These are other key subjects to be addressed when implementing an OTP token. we introduce a mobile cloud based otp token using REAL encryption algorithm [4] as the base cipher. The rest of this paper is organized as follows: section II presentsrelated work. The proposed

Recebido: dia/mês/ano Aceito: dia/mês/ano

authentication system provided in Section III. In Section IV, We Analysis Security of OTP Token and in Section V we analysis the security again attacks. Finally conclude our paper in section VI.

2 Background and related works

In spite a large range of services offered by cloud, most of the business organizations are not interested to adopt such services due to risks associated with security and privacy. The execution of the cloud subscribers' jobs in a trusted mode and assurance of the confidentiality, integrity, authenticity [5],[6] of uploaded data on cloud may increase the number of potential cloud's subscribers. The research organizations and academia are continuously identifying and resolving the security and privacy issues in cloud environment. However, there are still a few grey areas that must be addressed. In cloud the ownership and management of the users' data are separated, which cause that the worries of users to their own information resource become the important obstacle for the popularization of the mobile cloud computing. In addition the user's data are stored randomly in the shared infrastructure all over the world, and users do not know the specific position in which their data are stored. The loss of the physical control on the uploaded files laid foundation for new security issues that are identified and covered in [7],[8]. [9] list Eventual goal and indeed one of the major challenges of secured mobile could computing with privacy preserving ability is to provide reliable AAA (authentication, authorization and accounting) with different levels of access control (rule based or role based).

2.1 latest authentication schemes in mobile cloud computing

Slawmir et al in [10] have presented a mutual and two-factor authentication and claimed to be more secure against various phishing attempts than existing trusted third party protocols. [11],[12] used identity based encryption scheme for authentication users in mobile cloud. Many of the concerns and fear of the cloud has been addressed by Chow et al [13]. Instead of protecting data from outside using systems and applications, authors propose a self protecting data by injecting intelligence into the data to protect itself. Data needs to be self defending, self-describing and encrypted and packaged as per the policy without any dependence on the environment where it is stored or processed. Bin Liu et al [14] focused on privacy-preserving collaborative learning for the mobile setting to address supporting complex classification methods like support vector machines, respecting mobile computing and communication constraints, and enabling userdetermined privacy levels. [15] introduce a flexible framework for authorization decisions And uses implicit authentication. [16] used a Multiple token authentication strategy for mobile device. [17] introduce a Multi-modal biometric authentication algorithm with the ear fingerprint image.

2.2 Related work on mobile OTP tokens

Several implementations have emerged as the key methods to use a cellular phone in remote network authentication. One such solution focuses on using cellular phone as a standalone OTP token [18],[19]. The phone is a computational platform to generate the OTP code. The OTP generating software, user's secret seed and counter value are stored in the cellular phone. In operating the token, the user activates the software. The phone generates the OTP code. The user reads the OTP code and enters it into a PC or Internet device for 2FA need. Once authenticated, the user is allowed to access the network. these simple mobile tokens usually do not have any capability to resist the OTP seed (K) tracing by MITM interception or Shoulder-surfing attacks. Moreover, it stores the secret seed and counter value. these secrecies can be exposed if the phone is lost or stolen. The network security is comprised then.

An alternative proposal focuses on using the cellular network as a secure out-of-band channel to transmit or receive the OTP code to and/or from the authentication server [20]. The OTP code is generated by an OTP server. The OTP code is transmitted as an image data or text message via cellular network's Short Message Service (SMS) to the user's cell phone. The user then enters such OTP code to the authentication server to complete the 2FA procedure. Once authenticated, the user is allowed to access the network. An improved version [21] uses the

Recebido: dia/mês/ano Aceito: dia/mês/ano

mobile phone as the OTP receiver. The OTP code will be sent back to sever through mobile phone as well. So the OTP path and data path are completely separated. It completely prevents the traditional MITM attack in the network. But the Shoulder-surfing attack is still an issue. The cellular service providers cannot guarantee a real-time or in-time delivery, These factors limit the use of such OTP token.

The most recent proposal [22],[23] can be called a Mobile Authenticator. It involves using the Subscriber Identity Module (SIM) and other newly proposed communication protocols such as the Liberty Alliance Federation Standard [24]. The mobile device contains the user's credential information in its SIM card. The authentication is carried out directly through the phone, cellular network and the cellular service provider's server. Following the protocol procedure, once authenticated, the cellular service provider notifies the web server to allow the user's access. Two separate paths are used in this scheme. The user's credential data is only sent through the cellular network. The path separation prevents the MITM attack in the network. But the effectiveness of such scheme is limited by cellular service coverage. Table1 summarizes the properties of the various solutions of the mobile OTP tokens.

The key issue is that we need to resolve the security issues associated with this standalone mobile OTP token such as the protection of local stored confidential data. This is exactly what the proposed Mobile OTP Token sets out to do.

2.3 A review of the Rubbing Encryption Algorithm (REAL)

The REAL's goal was: 1) to securely encrypt the short OTP codes, 2) to safely deliver the ciphertext to a remote device through the Cloud, and 3) to securely store the ciphertext in a designated remote device. The remote device can be a personal computer, a notebook PC, a cellular phone, a mobile Internet device or other similar function products.

2.3.1 REAL encryption procedure

During encrypting, REAL places the original OTP code symbols as part of the elements of the matrix(X). Such matrix is also called REAL Image (RI). For ease of discussion, an M×N matrix X (REAL Image) and numerals of 0 to 9 as the set of symbol are chosen to illustrate the proposed REAL. A 6-digit (D=6) OTP code of "381269" is the plaintext for encryption. REAL encryption procedure is as follows.

1) Drafting a REAL image: REAL encryption starts out by designing a RI to be displayed on the remote device's screen. Figure 1 shows an example of REAL Image with size of 5×20.

2) Generating REAL key: REAL encryption key is the specific spatial locations (Wi) where the plaintext's symbols are placed inside the RI. Figure 1 shows those red circled locations where plaintext OTP code is hided.

3) Generating an actual REAL image: REAL places the plaintext symbols into the corresponding W_i locations in matrix X according to each symbol's occurring sequence. Since D=6, plaintext code=381269, we then have D5=3, D4=8, D3=1, D2=2, D1=6 and D0=9 for the plaintext code. In a RI with size T, each W_i has its own specific element a_{ij} in the matrix X. The rest locations are filled with randomly chosen symbols so that matrix X is equiprobable Such matrix X is the REAL Image.

Table 1 : Comparison among various types of Mobile OTP token

Category	Stand Alone	Token Out of Band	Mobile Authenticator
Role of phone	Computational platform	Transceiver of OTP code	Part of the authenticator
OTP generated by	Phone	Server	Phone and Server
OTP submission	Through PC	Phone, SIM & SMS	Phone, SIM & Protocol
Simple usability	Yes	No	No
No cellular limitation	Yes	No	No
Compatibility	Yes	No	No
Protect secrets	No	Yes	Yes/No
MITM attact safe	No	No	Yes
Shoulder-surfing attact safe	No	No	-

Recebido: dia/mês/ano Aceito: dia/mês/ano

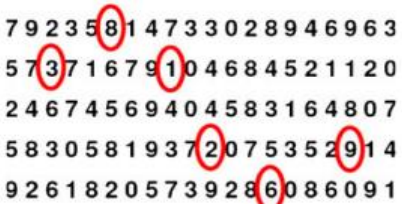

Figure 1. Example of a basic REAL Image (REAL matrix)

2.3.2 *REAL decryption procedure*

To decrypt the ciphertext, a user excute software token. Each Software token include of 2 key that points to key locations. The original OTP code is rubbed (decrypted) by the token and is ready for 2FA application. Notice that the user does not need to enter the REAL key to decrypt the REAL Image (ciphertext).

3 Proposed mobile cloud based OTP token

The REAL Mobile OTP Token is a mobile phone based standalone token using REAL as the base cipher. The token is Oauth compatible with 6-digit OTP passcode. It usesevent-based OTP algorithm (HOTP) [25]. It generates OTP code automatically without the aid of any PC or other Internet device. Using a mobile phone as the OTP generation platform presents itself certain difficulties. Mobile phone runs on a small battery. Mobile phone CPU is not as powerful as that used in the desktop PC. We need to reduce or eliminate the heavy number crunching operation as much as possible. The token contains certain secrecies inside the phone. As phone can easily get lost or stolen, securely protecting the secrecies becomes very important as well. So balancing a security level to trade off among the power,CPU and conformance is one of the most critical subjects when implementing a mobile OTP token. To do so, we do not compute OTP code directly on the phone set. All the OTP codes are pre-generated and encrypted using the low power encryption cipher such as Rubbing Encryption Algorithm (REAL) by OTP server. The encrypted codes is then stored in the designated phone. The user can then rub (decrypt) the OTP code using a REAL hardware token.

3.1 Designing software key

Software key contains pointers that points to key location on RI. Each software has tow key, the user by considering the RI select one of them. For 6-digit (D) of plaintext, we use 7 (= D + 1) code pointers as REAL key.

3.2 Encryption procedure

For ease of discussion, a REAL Image (X) of size 40 (T) and a set of 10 numerals of 0 to 9 are chosen to illustrate the proposed implementation method. The flow chart of REAL OTP operation if shown in Fig.2

3.2.1 *Generating REAL image*

In the actual provisioning process after a user activates her token, the server generates a series of OTP codes.

These codes are encrypted using the user's REAL key (embedded on her hardware token) to generate a series of corresponding REAL Images.

1) HOTP uses HMAC-SHA-1 hash function [26] for generating OTP codes.

$$OTP(i)=truncate(HMAC\text{-}SHA\_1(k,i)) \quad (1)$$

K is a 160-bit randomly chosen OTP seed and i is an event counter value. OTP seed K is a user specific random number. The event counter value (i) keeps the sequential number count of the OTP generation. It automatically increases by

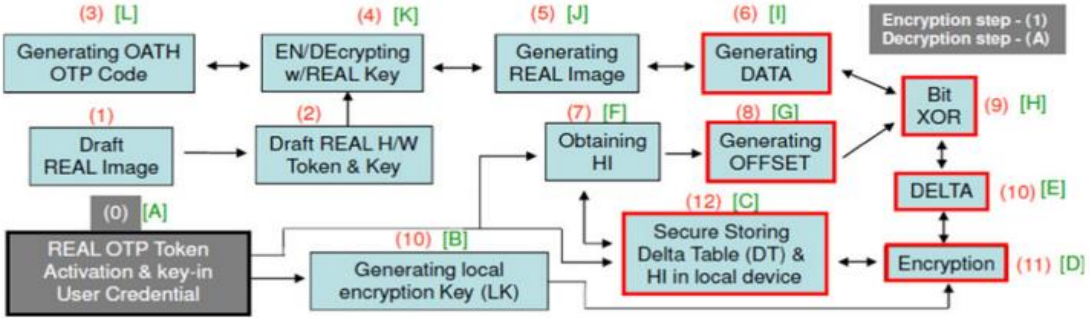

Figure 2 : Operating procedure of REAL mobile cloud OTP token

Recebido: dia/mês/ano Aceito: dia/mês/ano

one count after each OTP generation. SHA-1 hashes the two secrets to generate a 160 bit digest. HOTP then uses a dynamic truncation program to reduce the bit length to the range of 48 bit to 64 bit code or a 6 to 8 digit OTP code. The SHA-1 hash function guarantees the unique and irreversible of the digest.

2) Sequentially placing D_5 through D_0 into the corresponding W_i locations of the REAL image.

3) Fill an odd random number in W6 to indicate using number1 key. If number2 key is preferred, an even number will be filled in at W6 location.

4) Fill the rest of REAL image elements with randomly chosen symbols so that the image reachs equiprobable state.

5) Data(i) generate by the concatenation of all the elements of the REAL Image(i) (Fig. 3)

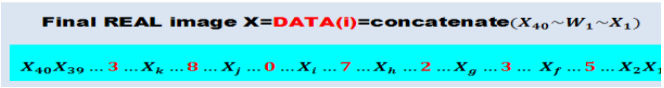

Figure 3 : REAL_Image(i) and Data(i) generation

To make up a 5000 units of OTP codes and REAL Images, we simply run the event counter value (i) from 1 through 5000. Then these OTP code send from server to mobile device.

3.2.2 *Offset generation*

To further protect the ciphertext data, we use a random value (Offset) to generate a logic operation difference (Delta) between Data(i) and Offset(i). Offset(i) is generated by a one-way hashing operation from the (i-1)th index value. This hashed index value is called HI.

$$HI(I)=HMAC\text{-}SHA\text{-}1(HI(i\text{-}1),1) \quad (2)$$

The initial hashed index HI(0) is determined by the following equation.

$$HI(0)=HMAC\text{-}SHA\text{-}1(K\text{-}1,1) \quad (3)$$

Once HI(i) is available, Offset(i) can be generated by taking the HMAC-SHA-1 operation to generate two 160 bit data (Fig. 4). The two 160 bit data are concatenated to form the Offset(i).

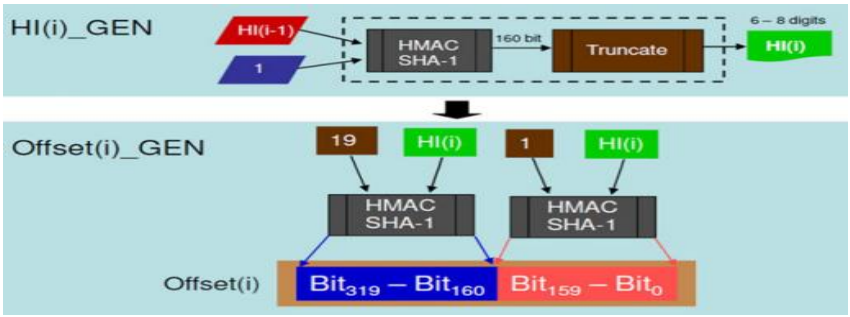

Figure 4 : Generating hashed index and offset

3.2.3 *Delta table and secure storage*

Delta(i) is the bit-Exclusive-OR (B-XOR) of the corresponding Data(i) and Offset(i) (Fig. 5). Their relationship is as

$$Delta_i=B\text{-}XOR(Data_i, offset_i) \quad (4)$$

Delta Table (DT) is the compilation of the entire Delta(i) in a special relationship to the corresponding HI(i). That is, Delta(i) is not stored according to the original sequence of Data(i). Delta(i) is rearranged to follow the value significance of its corresponding HI(i). The newly tabulated Delta(i) forms the DT. DT and the last HI data are then further encrypted using the algorithm similar to [27] "Scramble All and Encrypt Small" with the user's personal encryption key (CK). The encrypted data are securely stored in the designated mobile phone (Fig. 6). Personal encryption key (CK) is the concatenation of the userID and Mobile OTP Token password (MOTP).

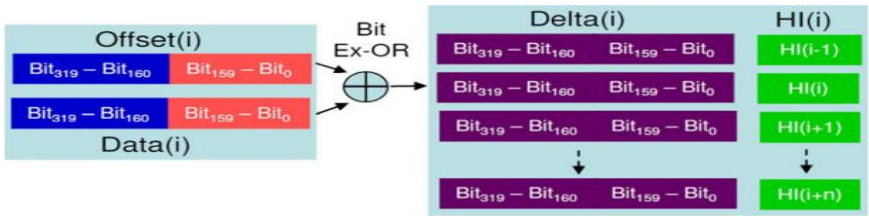

Figure 5 : Delta(i) generating

Recebido: dia/mês/ano Aceito: dia/mês/ano

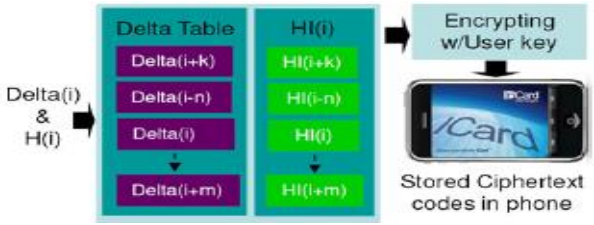

Figure 6: Generating Delta_Table (DT) and storing the DT inside mobile phone

3.3 REAL decryption procedure

The user activates OTP generation program. She then keys in her userID and password (MOTP). Personal encryption key (CK) is generated to decrypt DT and HI(i-1) stored in the mobile phone and, a new HI(i) value is generated using (2). Once HI(i) is available, Delta(i) can be found by sorting through Delta Table (DT). Offset(i) is also generated from HI(i). Data(i) is then obtained following (4). Subsequently, REAL_Image(i) is reconfigured from Data(i) and displayed on the mobile phone's screen. By running the software token, pointers are placed on the key locations. During the decryption process, the user always starts with first key (number1). TheOTP code rubbing sequence starts from the most top left symbol on the outer ring of the REAL Image (RI). The first symbol pointed by the first code pointer determines which key to use.

3.4 REAL mobile OTP client software

First, the user calls the server to initiate the OTP provisioning work. Server asks for the user's credential and token's serial number. It then associates the token's REAL key with the user's credential. The OTP server generates a series of 5000 OTP codes and randomly uses the user's REAL keys (2 keys per token) to encrypt the codes. Subsequently, the server generates the DT and HI and encrypted them with user's personal key (CK) to form the Data File. The encrypted Data File and Program File are zipped together at the end of the provisioning work. The zipped file can then be downloaded into the user's mobile phone through a secured Internet connection. After the auto-installation, both the Program File and the encrypted confidential Data File are securely stored in the mobile phone and ready for use.

4 Security analysis of OTP code and real image

For 5000 units of OTP code stored in a REAL Delta Table, its code size is about 200 KBytes. It is sufficient for a consecutive 2.7 years use with an average daily usage of 5 OTP codes. The entire software code size of REAL Mobile OTP Token is about 3.5 MBytes. REAL encryption is a fast and simple secure scramble cipher. It does not use complicated mathematical algorithm. It does not demand very fast CPU power either. So it can work easily with the less powerful CPU used in the mobile phone.

REAL Image (RI) contains the OTP code. Given an image size of 40 (T) and a 6-digit (D) OTP code:

$$\textit{Total number of REAL key} = C(40,(6+1)) \times 2 = 3.72E+0.7 \quad (5)$$

Without the aid of either a hardware token or a known OTP code, to correctly guess the REAL key from a RI, the probability (P_1) is

$$P_1 = \frac{1}{C(40,(6+1)\times 2}} = 2.7E\text{-}0.8 \quad (6)$$

This probability is again smaller than a straight brute force guess of the 6-digit OTP code. So even when a REAL Image is known to a malicious person, without the hardware token, a Mobile OTP token still stands very secure in such adverse situation.

5 Security attack safe real otp tokens

In this Section, we discuss and analyze certain adversary security attacks beyond the traditional MITM Replay attack. Two such attacks MITM Seed-tracing Attack and Shoulder-surfing Attack, are discussed and analyzed.

5.1 OTP token that resists MITM seed tracing attack

Seed-tracing is one form of the direct attack on an encryption algorithm (Fig. 7) For OTP token, the encryption algorithm and its implementation scheme must be well disclosed to the public. The attacker can take such disclosures and use certain mathematical mean to reduce the complexity of the algorithm. He then uses certain captured ciphertext to reversely trace the encryption key (seed). Some may even take certain pre-generated ciphertext (using the

Recebido: dia/mês/ano Aceito: dia/mês/ano

same algorithm and a known key) and compare to the captured ciphertext. Through these efforts, he then tries to find the pseudo random sequence of the captured samples. The secret encryption key (seed) may become reversible and eventually be found. A traditional OTP token does not have any built-in capability to effectively resist such attack.

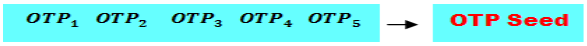

Figure 7: Tracing the seed by using OTP codes' pseudo random sequence

proposed REAL OTP Token can provide extra protection when MITM Seed-tracing attack happens. Each software token has two sets of REAL keys. The user can use one of the tokens in a mixed randomly order determined by the OTP provisioning server. Figure 8 shows such mixed random sequence generated by provisioning server. OTP_{Ai} and OTP_{Bi} mean the i'th OTP code generated by using seed A & seed B respectively. MITM intercepted OTP codes cannot show any viable pseudo random sequence data. It makes random number seed tracing very difficult. Such MITM Seed-tracing attack is then prevented.

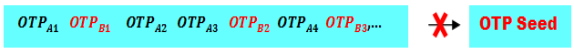

Figure 8 : Operation of MITM-seeding safe OTP token

5.2 OTP token that resists shoulder-surfing attack

Shoulder-surfing attack happens when a malicious person secretly observes the action while a user logins a destined server. The malicious person may then obtain the user credential information to trace the login secrecies or OTP code. Such attack does not involve with any mathematical cryptanalysis at the beginning. Once the secret information is collected, a cryptanalysis can be launched to find the next OTP code or trace the seed if many OTP codes are captured. Such attack does not happen in the electronic network.

It does not make any noticeable noise either. So it is very difficult for a user to prevent such attack. Almost all of the prevailing OTP tokens do not have any Shoulder-surfing prevention capability. To fend off Shoulder-surfing attack, we use a random offset concept in our REAL OTP token. The idea is to decouple the direct relationship between the code pointer location (REAL key) and the OTP code symbol. So the intruder can no longer find any meaningful data from the information captured through the Shoulder-surfing attack. As we said, D_6 determinates that use 1 or 2 key, we use D_6 will be used as a value added to each symbol's numeric value pointed by the rest of the six code pointers. The ten's digit will be dropped if the added value is greater than or equal to 10. The general equation is as follows.

$$D_i = (D_6 + D_i) mod\ 10 \qquad (7)$$

When making the REAL Image, each D_i value should be considered with the original OTP code according to (7). The table in Fig. 9 shows the entire OTP code decryption procedure. Symbols pointed by the code pointers are "807235" with D_6 equals 3. They are recorded in row 1 of this table. After adding D_6's value (3), the result is shown in the 2 row. We then drop the ten's digit (taking the mod 10 operation). The third row shows the final result which is the valid OTP code. Following this new decryption procedure, we have the full OTP code as "130568".

The modular operation also adds additional randomness to the codes. Such token then protects the code security and resists the attack from the troublesome Shoulder-surfing technique.

	D_6	D_5	D_4	D_3	D_2	D_1	D_0
I	6	4	7	8	8	1	8
II	6	10	13	14	14	7	14
mod 10	6	0	3	4	4	7	4

Figure 9: shoulder-surfing attack safe otp token

6 Conclusion

In this paper we presented an authentication mechanism in mobile cloud computing with combining the tow factor authentication and one time dynamic password called OTP token that suitable for mobile device. proposed token is based on Oauth standard and Compatible with other authentication system. Encryption operation is based on REAL encryption algorithm. It dont require any computation to decrypt otp code,so its Implementable in limit mobile device. as we shown is a strong authentication system and well resist against seed tracing, MITM-seed tracing and shoulder surfing attacks.

Recebido: dia/mês/ano Aceito: dia/mês/ano

References

H.Wang, S.Wu, M.Chen, W.Wang, (2014) ”Security protection between users and the mobile media cloud”, Communications Magazine, IEEE .

B.inarayanan L.aghavan, (2013) “Two-Factor Authentication”,springer, Apress.

Initiative for Open AuTHentication (2009) ,http://www.openauthentication.org/abou.

Cheng F (2010), “A Novel Rubbing Encryption Algorithm and The Implementation of a Web-based One-time Password Token.” In Proceedings of the 2010 IEEE 34th Annual Computer Software and Applications Conference.

Jia-Lun Tsai, (2015) " A Privacy-Aware Authentication Scheme for Distributed Mobile Cloud Computing Services”, Systems Journal, IEEE , Volume:9 Issue:3.

Gourkhede, M.H and Theng, D.P (2014) "Preserving Privacy and Illegal Content Distribution for Cloud Environment" International Journal of Computing and Technology (IJCT).

Debasish Jana, Debasis Bandyopadhyay (2013) ”Efficient Management of Security and Privacy Issues in Mobile Cloud Environment”, Annual IEEE India Conference (INDICON).

S.Vikas and K.Gurudatt (2014) “Mobile Cloud Computing: Security Threats”, International Conference on Electronics and Communication Systems (lCECS).

D.Dev and K.Baishnab (2014) ” A Review and Research towards Mobile Cloud Computing” 2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering.

G.Slawmir;p. Corcoran (2011) "Security analysis of authentication protocols for next-generation mobile and CE cloud services" International Conference on Consumer Electronics - Berlin (ICCE-Berlin)

Dijiang Huang, X. Z (2010) "MobiCloud: Building Secure Cloud Framework for Mobile Computing " IEEE International Symposium on Service Oriented System Engineering , 27-34

D.Huang, Z. Z. (2011)" Secure Data Processing Framework for Mobile Cloud Computing". IEEE INFOCOM workshop on cloud computing , 614-618.

R.Chow, P.Golle, M.Jakobsson, E.Shi, J.Staddon, R.Masuoka, and J.Molina. (2009). “Controlling data in the cloud: outsourcing computation without outsourcing control.” In Proceedings of the ACM workshop on Cloud computing security (CCSW '09).

B.Liu, Y.Jiang, F.Sha, and R.Govindan. (2012).”Cloudenabled privacy-preserving collaborative learning for mobile sensing.” In Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems (SenSys '12).

Richard Chow, Markus Jakobsson,(2010) ”Authentication in the Clouds: A Framework and its Application to Mobile Users”,ACM.

A.Ahmad, M. M. Hassan, A. Aziz,(2014)” A Multi-Token Authorization Strategy for Secure Mobile Cloud Computing” , 2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering

A. Tharwat1, Abdelhameed F. Ibrahim2, Hesham A. Ali(2014)” Multimodal Biometric Authentication Algorithm Using Ear and Finger Knuckle Images ”

Aloul F, Zahidi S, El-Hajj W (2009) “Two Factor Authentication Using Mobile Phones.” In Proceedings of the 2009 IEEE/ACS International Conference on Computer Systems and Applications.

Deepnet Security (2010) MobileID - A Mobile, Two-way and Twofactor Authentication. http://www.deepnetsecurity.com/products2/MobileID.asp

Liao K, Sung M, Lee W, Lin T (2009) “A one-time password scheme with QR-Code based on mobile phone.” In Proceedings of the INC, IMS and IDC, 2009. NCM '09. Fifth International Joint Conference.

S.H Park; S.Hyung Lee (2012) “Token-Based Mutual Exclusion Algorithm in Mobile Cellular Networks” IEEE 26th International Conference on Advanced Information Networking and Applications Workshops (WAINA)

Recebido: dia/mês/ano Aceito: dia/mês/ano

Thanh D, Jonvik T, Thuan D, Jorstad I (2009) "Strong authentication with mobile phone as security token." In Proceedings of the IEEE 6th International Conference on Mobile Adhoc

Z.Su , Q.ianhua He , J.Zhang , H.Li,(2013) »Research of Single Sign-On in Mobile RFID Middleware Based on Dynamic Tokens and WMMP«, IEEE 16th International Conference on Computational Science and Engineering (CSE).

Liberty Alliance (2010) Liberty alliance project. http://www.projectliberty.org/

M.Raihi D, Bellare M, Hoornaert F, Naccache D, Ranen O (2005) "HOTP: An HMAC-Based One-time Password Algorithm". The Internet Society, Network Working Group. RFC4226

Eastlake D 3rd, Jones P (2001) "Secure Hash Algorithm 1 (SHA1)". Network Working Group. The Internet Society (2001).

Mizuno S, Yamada K, Takahashi K (2005) "Authentication Using Multiple Communication Channels". In Proceedings of the DIM 2005 Conference.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Recebido: dia/mês/ano Aceito: dia/mês/ano
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~