A Stable and Secure One-Time-Password Generation Mechanism

Article *in* Journal of Advanced Research in Dynamical and Control Systems · June 2019

CITATIONS
3

READS
1,258

2 authors:

Sajaad Ahmed
Islamic University of Science and Technology
26 PUBLICATIONS **136** CITATIONS
SEE PROFILE

Ajaz Hussain Mir
National Institute of Technology Srinagar
67 PUBLICATIONS **388** CITATIONS
SEE PROFILE

A Stable and Secure One-Time-Password Generation Mechanism

Sajaad Ahmed Lone, National Institute of Technology, Srinagar, Jammu and Kashmir, India.

E-mail: lonesajaad786@gmail.com

A.H. Mir, National Institute of Technology, Srinagar, Jammu and Kashmir, India. E-mail: ahmir@rediffmail.com

Abstract--- Security concerns are rising today in all domains, such as banks, governments, medical assistance, military organizations, etc. Several solutions have been proposed but security loopholes and user flexibility still remain a big question. The past decade has seen a growing interest in replacing username-password authentication with new methods. On the other hand, there is a tremendous growth in web-based applications. Two factors are considered key elements in the usage of password authentication: usability and security. The authentication systems based on One-Time-Passwords (OTPs) are the most commonly used. In this paper, various OTP generation mechanisms have been reviewed and a novel OTP generation method has been presented. The OTP produced from the system is user-friendly as well as secure since it is a 6-digit OTP generated using RIPEMD160.

Keywords--- One-Time-Password, Security, Authentication, OTP Generation, Insert.

I. Introduction

One of the interesting facts in the area of network and security is that with the prominent evolutions of new tools and technologies to mitigate security threats. There is also an exponential rise in attacker's policy to intrude and invoke various malicious activities over the secured network. Every year, news on security breaches evidently persists where maximum attacks are targeted towards stealing the authentication rights of a legitimate user. Usually, the intruder seems to exhibit a typical behavior over the internet. They keep a track of potential victim, surveil their online behaviour, understand their browsing pattern and then create a hit list of similar types of victims. In order to do so, there are already software's and various types of malicious codes which never seemed to be thwarted till date. Hence, the outcome of such malicious activities cost the sensitive and confidential transactional data of legitimate user finally resulting in identity theft. Such types of intrusion activity have its collateral effect on multiple database and web-server affecting millions of users online.

The security breaches over authentication as well as authorization policy were always on rise till date and therefore, it has attracted numerous researchers to find out some sort of robust and fail-proof solution. Henceforth, in the advent of a massive volume of research work previously addressing such issues, interesting and uniquely, it was found that there exist a unique group of studies that focus on using multiple parameters to be considered at the time of authenticating a legitimate user.

In information security, one of the main concerns is the verification of an individual demanding access to confidential, classified and sensitive information as an authorized one. This can be achieved when that individual proves his identity by means of an authentication process. In other words, the individual should be able to justify his identity in order to access information and in case he fails to authenticate himself, access will not be granted. In general, an authorized user can be identified in three ways viz. what an individual knows, what an individual has, or what an individual is. Among the three, the usage of what a person knows i.e. passwords etc. is the most common. The usage of what a person has i.e. One-Time-Password (OTP), tokens, smart cards, etc. is used for sophisticated authentication. The third method, what a person is, involves biometric technology [1].

It has been observed that the usage of former two security schemes, which is called two-factor authentication, is not enough. Therefore, a third higher degree of authentication can be augmented to enhance the authentication strength by taking into consideration something the individual is. Identification of users via biometric features can be performed on the basis of behavioural or physiological characteristics. This implies utilizing some traits of the person that cannot be modified or mimicked easily, e.g. facial features, fingerprint, eyes, etc. As per the information provided by the International Biometrics Group, "There is no one right biometrics technology for every application"[2].Three-factor authentication is defined as the usage of three independent authentication mechanisms. Such an authentication level necessitates the utilization of passwords, tokens/smart cards and personal identifiers.

Received: 31 May 2019/Accepted: 18 June 2019

A hardware token or smart card might be needed by those who need to access a specific application rapidly. Those having minimum security requirements can use a password for authentication purposes.

The introduction of biometrics as the third level of authentication will undoubtedly enhance the efficiency of the authentication mechanism in comparison to the conventional usage of a password, tokens/smartcards or their combination.Nevertheless, there exist multiple points where biometric systems can be breached [3]. Some threats have been addressed by researchers in [4][5][6][7][8][9][10][11][12] still biometric systems remain susceptible to spoofing and smudge attacks. Even if there exist systems offering protection to biometric templates by revoking biometric credentials, such solutions have limited availability, and underdeveloped standards exist to evaluate such solutions. Biometric traits are not secretive; they are static in nature, one can take a picture of someone's face with or without their information, etc. Contrariwise, there exist solutions to safeguard against such threats in the form of Presentation Attack Detection (PAD) such as liveness detection [13].Unfortunately, biometric systems appear to be highly susceptible to replay attacks [14]. Hence, security solutions based on biometrics alone offer weaker security even if they provide high user adoptability.

Security modernisation in the current era still finds the use of single-factor or password-based approach for access management across digital channels as they offer affordable deployment, easy revocation in the event of a compromise. Existing digital application services require a multitude of credentials in the form of PINs and passwords to be remembered by individual users resulting in higher proclivity of unsafe password selection by users for easy memorisation, which proves to be as one of the considerable concerns associated with it [15]. However, in the current era, rapid modernisation and development in computing technology, susceptibility to dictionary attacks [16], copious key-logger [17] and password hacking tools are available [18][19], making password retrieval an easier task for intruders. Further, passwords can be shared, forgotten or observed thus forming an impractical authentication solution.

Currently, one time passwords are widely used for authentication and authorization due to the comfort ability, ease of use and higher adoption by the majority of users. Various OTP schemes are explained as follows:

1. **Time synchronized OTP:** It is usually related to some hardware referred to as a security token. Within the token, there is a precise clock which has been coordinated with the one on the registered authentication server where time is a significant factor for password generation because the new password generation is grounded on the present time, or beside, a secret key or the previous password.
2. **Lockstep synchronized OTP:** Each time a fresh password is produced on a device, it increments its own counter and each time a user tries to log in, the server will increment its internal counter. Typically, the next x passwords may be accepted, even though the things go a little bit out of synchronization, the server can automatically resynchronize it.
3. **SMS OTP:** This is a completely different approach where the server shall transmit a password to the user device during authentication through SMS. No seeds are required. And the passwords are completely random.
4. **Challenge- response OTP:** OTP is generated through the Server Challenge. It needs a user to deliver a response to a challenge during authentication.

II. Related Work

Various problems pertaining to authentication and security of highly privileged and private information have been analyzed by many scholars. While studying diverse schemes adopted in the present as well as the past, it was observed that the usage of One-Time Password or commonly known as OTP seemed to assure improved access management security in private as well as public networks [20].

Owing to the varied formats of OTP usage as well as the architecture developed by the past protocol makers and researchers, diverse OTP schemes have also been patented but standardization of the same is a challenge. Various OTP based schemes are discussed in this section. The security of the systems based on OTP relies on the non-invertible property of secure hash functions. This implies that the function can be computed in the forward direction, but its inverse calculation is computationally infeasible [21].

A. One Way Function Chain Approach to Generate OTP

It was Lamport who first recommended the idea of OTP in the initial years of 1980 [22]. Lamport proposed a popular OTP scheme based on a one-way hash function that warded off replay attacks in every communication.

Received: 31 May 2019/Accepted: 18 June 2019

The principle of OTP ensures improved security by using an algorithm that generates a pseudorandom output every time a user tried to log in. The password system in this scheme begins with the initial seed *s*, and proceeds with the generation of passwords via a one-way chain function, *f(s), f(f(s)), f(f(f(s)))*,….and so on. If the number of passwords required is unknown, a new value of seed may be selected after exhausting the set for *s*. For authentication, the user device provides the passwords in a reverse manner beginning from *f(f(f(s)))* to *f(s)*.The limitations of utilizing Lamport's scheme have been given below:

- The total number of authentications (N_A) permitted by the algorithm is static. In case more authentications are needed, then a new series of OTPs must be produced.
- The value of $N_A$should be so small that the computation of F^{N_A} is performed. For N_A OTPs required, N_A compositions of F need to be computed by the user. However, this process is time-consuming and needs extra storage for storing passwords, e.g., for N_A= 220, the user needs to compute $F(2^{20})$ that requires greater than 1,00,000 compositions and space or time might become an issue in case of limited computational resources.
- For an indefinite number of authentications, the user might not actually require N_A OTPs, e.g., if any user desires to authenticate for N_A= 220 for a year but actually authenticates just ten times, then a great deal of computation becomes worthless.

B. S/keyTM One Time Password

The author in [23] a prototype software called the S/KEY™ OTP authentication scheme that employs a computation for generating a definite series of one-use passwords out of a solitary secret 'seed'. The security of the system relies on the seed that is known to the user only. This system is an extended version of Lamport's scheme. The one-time passwords are so related that computation of any password from the preceding sequence is computationally infeasible. The process involves the application of a hash function $h(.)$ to the seed s for N instances forming a hash chain with the length equivalent to$N$, as

$$h^1(s), h^2(s), \dots\dots\dots, h^{N-1}(s), h^N(s) \quad (1)$$

At t[th] instance of authentication, the user receives a challenge from the host:

$$Challenge\ (t) = N - t \quad (2)$$

Then, t[th] OTP is computed by the user as per the received challenge as

$$OTP_1(s) = h^{N-t}(s) \quad (3)$$

The host then checks the authenticity of the user as per the condition:

$$h\left(OTP_1(s)\right) = h^{N-t+1}(s) \quad (4)$$

In (4), the value of $h^{N-t+1}(s)$ is already stored in the password file of the host system from the last $(t-1)^{th}$authentication. After every successful authentication attempt, the password file with the user is updated with the password stored before the final hash computation of$h^{N-1}(s)$. In such a case, $t$is incremented by one by the host who then sends a fresh challenge to the user for subsequent authentication. However, this system is restricted to N authentications only, i.e. the system has to be restarted after N authentications. Besides, this system is vulnerable in case the attacker impersonating the host sends a challenge of some small value to the user who replies with the initial values of hash chain allowing the attacker to compute more OTPs [24]. Such an attack is called 'small challenge' attack. Furthermore, during the computations for the initial values of the chain, the computational requirements of the user are higher making the system infeasible for limited resource devices i.e., mobile phones. Even though the scheme is immune to eavesdropping and replay attacks, it has been found to be susceptible to Server-spoofing attacks as well as offline dictionary attacks.

C. Bicakci et al.'s Scheme

Authors in [25] proposed Infinite Length Hash Chains (ILHC) that employ a public key cryptographic algorithm, say A for producing an infinite and forward one-way-function or OWF that forms the production core of OTPs. This system has employed RSA [26] with d as private key and e as the public key. The one-time-password obtained after applying the RSA algorithm to the first input s for n[th] authentication is given below:

Received: 31 May 2019/Accepted: 18 June 2019

$$OTP_1(s) = A(s,d) \quad (5)$$

And the n[th] OTP is verified by the decryption of $OTP_1(s)$ with the help of e as:

$$A(OTP_1(s), e) = OTP_{n-1}(s) \quad (6)$$

But the increase in the cascaded exponentiations raises the computational complexity thus making its implementation difficult in devices with restricted computation like mobile phones.

D. RSA SecureID Authenticator

Hardware tokens are employed for storing secrets in dedicated modules carried by users. The RSA token family [27] has long established as the market leader. Here, a simple dedicated hardware version has been employed which has just a display without any I/O ports or buttons. The device holds a secret referred to as 'seed' that is known at the backend and is synchronized with the main server's internal clock. A cryptographically strong algorithm will generate a fresh digit code for 60 seconds utilizing the seed and current time stamp. The generated code is displayed on the device screen.

When enrolling, the user uses a web interface to connect to the administrative backend where he/she chooses a PIN and the username-token pairing is affirmed. After this, the user types his/her 'username' and 'passcode' (the concatenated form of a dynamic 6-digit code and a static PIN of4 digits) in place of username and password for authentication. A similar process occurs at the server end parallelly. RSA provides a single sign-on facility for granting access to various corporate resources using the same token. However, it was found in March 2011 that attackers corrupted the backend seed database of RSA [28] allowing them to guess the codes supplied by any token. Owing to the typical nature of mobile devices, this type of synchronization cannot be assured.

E. Unidirectional OTP Scheme

In this scheme, the client as well as the server shall generally have the same counter. The client produces an OTP from a secret key counter value and then increments the counter value. The user submits this OTP to the server and server also produces a password by making use of its counter value and the same secret key that the user employed. In the case of a password-match case, the user gets authenticated by the server and its counter value is incremented. The HOTP scheme [29] is unidirectional that relies on a counter value and utilizes the HMAC-SHA1 algorithm for OTP generation. This computation yields a 160-bit output that is truncated by HOTP to a form that can be typed by the user easily.

$$HOTP(K,C) = Truncate(HMACSHA1(K,C)) \quad (7)$$

Here, HMAC-SHA1 is a unidirectional hashing function yielding 160-bit output.

Truncate signifies a function generates an OTP from HMAC-SHA1 value.

C represents the value of an 8-byte counter that should be synchronized between the HOTP validator (i.e. server) and HOTP generator (i.e. client).

K represents the secret shared between the server and the client; every HOTP generator possesses an inimitable secret K.

The counter value of the server only raises when an authentication is successful, however, the user's counter value raises at every request of a fresh OTP. Thus, the counter value of the server and the user might be distinct, referred to as counter desynchronization. In case of counter desynchronization, the counter value of the server is raised first and a match of the user's password with the new one (computed from the counter value of server and secret key) is sought. The server carries on this process until a specific threshold T is reached, T being called the look-ahead window, thus concluding that the password is not valid. Nevertheless, the value of T is based on experience and since the server expends much time on computation, it cannot be large enough which may lead to DoS attack if an illegal password is submitted by an attacker. In case its value is small, a genuine user may not be authenticated with a legal password when the counter difference between the server and the client gets greater than$T$.

F. OTP Generated by MID Let

A new, secure authentication mechanism has been given in [30] that relies on the principle of one-time-passwords. This OTP solution brings together the secure and simple OTP principle pervasiveness of a GSM mobile device. A Java MIDlet transforms the mobile phone into a safe OTP token since it can be installed on every Java enabled phone which can then be utilized for logging into any web service.

Received: 31 May 2019/Accepted: 18 June 2019

This solution relies on the plain challenge-response mechanism. For making the OTP generation secure, a tough secret key is employed in concatenation with a challenge. The secret key must be shared between the server and the client.

An enhanced version of the Diffie-Hellman key exchange, SPEKE (SPEKE (Simple Password Exponential Key Exchange) [31] has been employed for exchanging the key securely. This key exchange between java MIDlet and Java Servlet happens through SMS over the GSM network. To avert the intruder from knowing the secret key, it is encrypted using the extended version of the password chosen by the user as the encryption key before its storage on the device. The hash of the password of the user is also stored such that the password can be verified at the start up by the MIDlet. The AS stores secret key along with User profile in its database. The OTP generation is performed as:

$$OTP = hash\,(Challenge||secret\ key) \tag{8}$$

Whenever a user needs to be verified, Authentication Server (AS) generates a Challenge with the User profile and the corresponding OTP; then, it computes the MAC of OTP. The MAC, as well as the challenge, are transmitted to MIDlet in an SMS whereas the OTP is retained at the AS. The MIDlet is activated automatically at the arrival of the message and the user is prompted for a password. The MIDlet produces the password hash and verifies it against the value stored. If they match then it will be used to decrypt the shared secret key. MIDlet will also generate OTP and its MAC is calculated and compared with the received MAC. If it does not match, the authentication shall be cancelled, and the process should be restarted. If they match, then the user will be successfully authenticated. The MIDlet will then send OTP back to AS. The server checks if the OTP is right and authenticates the user.

G. Symmetric Encryption Algorithm and One Way Hash Function To Generate OTP

Authors in [32] and others proposed another OTP scheme which uses a one-way hashing function and symmetric encryption (AES-128) for OTP generation. The one-time-password say P is formed of two things, $P = C||D$, where C signifies cipher text and D represents verification bits. A cipher text of k bits can produce 2^k distinct OTPs. The length of verification bits relies on the security requirements. The output obtained from symmetric encryption carries the information of counter value to the server thus avoiding counter desynchronization. However, employing only symmetric encryption is insufficient for OTP generation. If a random password is submitted by the intruder, it is decrypted by the server that attains the counter value. It is highly likely that the value of the counter is greater than the present counter value saved at the server and may lead to a successful assault. Hence, verification bits are added to OTP which are formed from the counter value digest for ensuring its validity.

The proposed scheme works in 3 stages: key distribution, OTP generation and authentication. During key distribution phase, the server generates a key K_c for every user and then supplies a token with K_c and ID to the user. In the phase of OTP generation, the user device produces an OTP P in the following steps:

1. $INC(i)$
2. $C = E_{kc}\,(i)$
3. $D = Truncate\,(H(ID||K_c||i))$
4. $P = C||D$

On reception of the OTP P, the server checks if P valid using the below-given steps is:

1. Check if P is forged

For $P = C||D$, the server computes $i = D_{kc}\,(C)$

$$D' = Truncate\,(H(ID||K_c||i))$$

If $D' \neq D$, then the server concludes that D is forged and rejects the login request. Or else, the server continues to check P through succeeding steps.

2. Check if P has been used previously or not. Only if P is new and has not been submitted before, the authentication shall be successful. Otherwise, authentication will be failed. Update the currently stored password to new password on successful authentication.

Received: 31 May 2019/Accepted: 18 June 2019

3. When an authentication fails, the server logs the invalid submission to ward off DoS and guessing attacks. If the number of invalid user submissions reaches a specific threshold, the server restrains the user for subsequent submission.

H. Time-Based OTP

TOTP [33] is an extended version of HOTP for supporting a dynamic factor based on time. A dynamic factor is a value which should vary every time a fresh password is produced for ensuring the generation of different passwords. Basically, TOTP is defined as

$$TOTP = HOTP(K,T) \quad (9)$$

Here, T as an integer signifies the time-steps between the present Unix time and the initial counter time T_0 (Unix epoch). The OTP produced in one time-step shall remain the same. When the client sends an OTP to the host, the host is unaware of the actual timestamp of the OTP generation. In general, the host uses the timestamp when the OTP is received for evaluation. The time gap between arrival and generation may be large owing to network latency. The OTP generation time may fall to the end of a one-time stamp window and reception will fall within another timestamp window. Hence, the host will adopt a policy to set the transmission delay window of OTP for authentication. The authentication system should match the OTPs with the previous timestamps which lie in the transmission delay and not just with the receiving timestamps.

The size of the time-step affects both usability as well as security. The large size of time-step exposes a bigger window to assault. When the OTP produced is subjected to a third party, he/she can consume OTP within the time-step window. Another disadvantage is that the user shall have to wait until the clock reaches the subsequent time-step from the previous submission. A bigger time-step window implies that the user has to wait for a longer time for attaining the subsequent valid OTP after the previous successful OTP authentication. Even larger window shall not be feasible for conventional Internet login purposes.

Google 2-factor authenticator uses TOTP for the generation of OTP. Since all TOTP systems rely on the clock of User's phone to match the clock on Server. This Google's authenticator can go out of sync if there is no network access.

I. OTP Based Two Factor Authentication using Mobile Phones

The idea given by [22] was extended with some modifications for obtaining forwardness and infiniteness, evading the usage of public key cryptography. The disadvantages of those parameters lead to various susceptibilities presented with respect to [24]. The Lamport's Scheme was integrated with two distinct hash functions, h_A and h_B, the second hash function h_B allows going into the forward direction by generating the chain produced by h_A as:

$$OTP(x,y) = h_B^y(h_A^x(seed)) \quad (10)$$

Authors in the study [34] have discussed the utilization of two-factor authentication using OTP. However, after in-depth scrutinizing the work of Eldefrawy, it is found that generated OTP is large enough (68606061177919188523363813602016) which is not user-friendly (it is difficult for the user to read the 32 digits & enter into the interface for authentication purpose). The transformation from digits, hashing the output to characters, the format of the password was not covered. The major security loopholes found in the usage of OTP is that it generates the secure password that floats into a GSM network where there is a higher degree of intrusion. Another prominent issue found in the majority of the OTP usage (as used by Eldefrawy) is that it is not preferred for mobile phones due to time-synchronization that are usually based on an internal clock synchronization system. Moreover, their study has deployed the hash function using SHA-1 and MD5 where there already exists the attack report on usage of SHA-1 and MD5 in the vulnerable public network.

III. Open Issues

After conducting the review of previous research works, the open issues explored have been given below:

- While exploring diverse mechanisms adopted in the existing as well as the past systems, it is observed that the employment of One-Time Password or commonly known as OTP seems to assure improved access management security in both private and public network [20]. While making a transaction, the OTP is valid for a single access attempt only. The usage of OTP obviously provides fail-proof security against replay attacks since the generated password is invalid for the next time and thus cannot be exploited by the attacker. As a result, the utilization of OTP has been scrutinized in this study for exploring better

Received: 31 May 2019/Accepted: 18 June 2019

possibilities to enhance system security for user authentication. Owing to the varied formats of OTP usage as well as the architecture developed by the past protocol makers and researchers, diverse OTP schemes have also been patented but standardization of the same is a challenge.

- Though numerous researchers have worked on enhancing the security of two-factor authentication systems, as observed in the previous section but the efficiency of those schemes has not yet been proved. Any malicious event dearly costs the user its system account. Despite the provision of highest security by the user, the user system might already be endangered. The research works investigated in this study have just focused on the primary verification level employing two-factor authentication that may not be used eventually after the successful verification of the user the first time. Particularly, the studies conducted by authors in [35] and [36] stress on the illicit intrusion attempts. It is thus evident that focusing on the basic access level for creating a strong security system cannot be dependable in the long run owing to various malicious programs that may compromise the security policies in near future and steal users' identities successfully. There is not a single study among those given in the previous section that stress on the escalation of security privileges where the chances of malicious interference are always the most and it roots from normal user access. Further, if such a malicious program incorporates into the system successfully, the user may not be ever able to know its root location for performing quarantine.
- Another possible scenario that has not been considered in the previous studies is the Man-in-the-Middle attack. This attack scenario employs an illicit proxy server located between the authentication server and the communication channel. At the arrival of a service request, after authentication token generation, the token passes through the unsafe routes leading the crucial information to the attacker. And when the information is stolen, the attacker may configure the whole authentication system easily and thereon, the attacker would have permanent access to resources. Such attack scenarios have not been observed to be considered in the previous works thus being an open issue.

IV. Proposed OTP Generation Mechanism

The purpose of this study is the generation of human-readable OTP on the mobile device that will be used for authentication purpose. One of the noteworthy enhancements that have been performed was ensuring a higher security level incorporation. The literature review conducted in this study discusses the utilization of hash functions such as SHA-1, MD5, etc. that are no longer considered secure algorithms in cryptography. SHA-1 proffers many concerns when implemented on a public network and thus should not be chosen (or should be modified) by an investigator when the experimentations are to be performed in large public networks with an advanced level of unguided intrusion events. As a result, the study adopts RIPEMD128 in place of the error-prone and traditional SHA-1 algorithm. In the cryptographic algorithm, working with digest is the most conspicuous phase in the design process. Thus, incidentally, exploring the work of Eldefrawy, it was found to use 128-bit digest to produce a 32 digit OTP which is not user-friendly because it is difficult for the user to read and enter 32 digits OTP into the system. The existing study by Eldefrawy might be a better standard for this study by discoursing the implementation of a security protocol; nevertheless, it was not user ergonomic. As the numeric format of OTP was found very complex and error-prone, therefore, enhancing this technique by converting 128-bit digest into six digits is done thus making the OTP long enough to be secure but short enough to be user-friendly.

The proposed system utilizes a combination of hardware/software profiles of the mobile device as the initial seed for generating a One-Time-Password.Every smartphone has Bluetooth and WiFi facilities, so concatenation of the duo shall form the hardware profile in our system. Froyo was launched in May 2010 and superseded by Gingerbread in Dec 2010. Since Gingerbread, all versions have a software identifier called "Serial number" that comprises the software profile in the proposed system. The added advantage it imparts is that also forms a part of other Android devices which lack call facility.

The seed generated from the previous step is fed to RIPMED160 for hash generation. Instead of 128-bit hashing and 128-bit random number, 160-bit hashing and 160-bit random number shall be employed. It shall serve two advantages: i) 160-bit hash are more secure ii) TOTP already standardised a Dynamic Truncation method of 160-bit hash values for deriving 6 digit OTP numbers. For 160 bit hashes, SHA 1 is the most popular, but attacks are existing at the theoretical level with SHA 1. RIPEMD 160 thus emerges as a better choice.

The next step in the OTP generation process as shown in Figure 1 involves a 160-bit random number generation that is XORed with the 160-bit message digest generated in the last step to yield a 160-bit pass.

Received: 31 May 2019/Accepted: 18 June 2019

Then, dynamic truncation is performed to convert this 160-bit pass into a 6 digit, human readable and user-friendly OTP. The OTP generated can be then transmitted using SMS, push message, e-mail, etc.

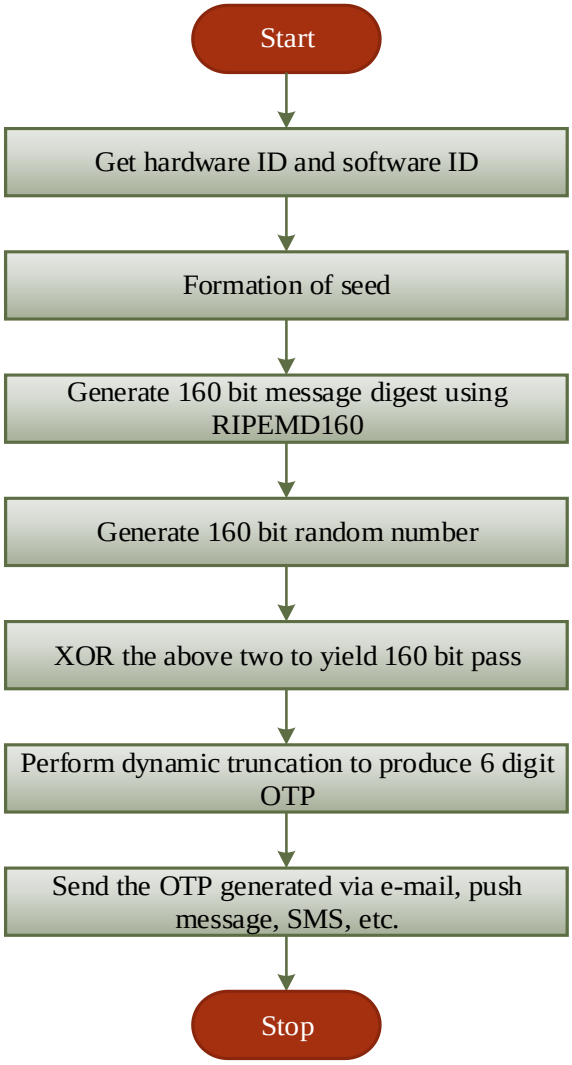

Fig. 1: Proposed OTP Generation Process

V. Security Analysis

OTP generators should resist most of commonly known attacks like replay attack, offline/online guessing attack, pre-imaging attack, brute force attack, dictionary attack, rainbow attack, man in the middle attack, birthday attack, forgery attack, full key attack, recovery attack, and collision attack but should not be limited only to these . In order to validate the proposed OTP generation method a simple security analysis is carried out with respect to the previously defined security threat.

In the proposed method RIPEMD-160 is used as a hash function which produce a strong 160 bit hash string from the Hardware and software ID.RIPEMD-160 is cryptographic hash functions recommended as a drop in replacement ofSHA-1. Even though RIPEMD-160 relies on the same design principles as MD5 and SHA-1, the dual-stream structure makesRIPEMD-160 more secure against recent attacks on other members of the MD4 family**.**

The proposed scheme will resist brute force attack as brute force does not depend on the specific algorithm but only on depends on the bit length of hash value. In the proposed OTP generation scheme RIPEMD-160 bit hash function is used to produce hash from software and hardware ID which is then XORed with 160 bit random number .So if an adversary wishes to find the hash value the level of effort will be proportional to 2^{160}.

Received: 31 May 2019/Accepted: 18 June 2019

In the worst case, an adversary has to exhaustively examine a search space of possible combination of which is equal to 2^{160} only to get hash value ,XOR operation and dynamic truncation of the hash adds more complexity for such an attack.

The proposed scheme is not vulnerable to reply attacks, man in middle and forged attacks as one time password generated only for one authorization or authentication request. Even if adversary will intercept valid OTP it can not be used in subsequent login as OTP is restricted to short time window. The proposed scheme also makes it difficult for adversary to generate a new OTP from last observed one because of huge computational cost.

VI. Conclusion

Today, not only mailing agents and banking sector need to worry about the security of their OTPs. All the companies from various sectors and sizes make use of OTP for authenticating their clients to access the available resources in a user-friendly way in the present cut-throat business environment. Therefore, there is a need to develop a user-friendly and secure OTP generation process and one such model has been presented in this study. This model generates OTP using the hardware and software profiles of the user device as the initial seed. In this system, RIPEMD160 has been employed as the hashing function and the end resultof the process is a 6-digit OTP. The OTP generated can be transmitted to the users by means of email, push message, etc.Although the basic design of our system can be rooted from the idea formulated by Eldefrawy, still it has some of the potential contributions of results and accomplishments which is quite unique in its nature. The base technique has used conventional One-Time password by means of two-factor authentication using the SHA1 algorithm. It has been strongly argued by NIST that currently, SHA1 is not the most potential cryptographic hash function. Therefore, our first contribution can be stated as incorporating a latest hash function RIPEMD128 in our system. Adopting this technique of enhancement will yield an OTP that is potentially strong compared to the basic approach.

References

[1] G. Jiwnani and N. Saxena, “Multi-modal Biometric Authentication using Fingerprint and Iris: a Review”, International Journal of Computer Science & Communication Networks, vol. 5, no. 2, pp. 115-119, 2015.

[2] A.K. Jain, R. Bolle and S. Pankanti, Biometrics: Personal identification in networked society. Kluwer Academic Publications, 1999.

[3] N.K. Ratha, J.H. Connell and R.M. Bolle, “An analysis of minutiae matching strength”, in 3rd International Conference on Audio-and Video-Based Biometric Person Authentication (AVBPA), Springer, Berlin, Heidelberg, 2001, pp. 223-228.

[4] P. Ambalakat, “Security of biometric authentication systems”, 21st Computer Science Seminar, 2005, pp. 1-7.

[5] G. Pan, L. Sun, Z. Wu and S. Lao, “Eyeblink-based anti-spoofing in face recognition from a generic webcamera”, IEEE 11th International Conference on Computer Vision, ICCV, IEEE, 2007, pp. 1-8.

[6] H.K. Jee, S.U. Jung and J.H. Yoo, “Liveness detection for embedded face recognition system”, International Journal of Biological and Medical Sciences, vol. 1, no. 4, pp. 235-238, 2006.

[7] W. Bao, H. Li, N. Li and W. Jiang, “A liveness detection method for face recognition based on optical flow field”, in International Conference on Image Analysis and Signal Processing, IASP, IEEE, 2009, pp. 233-236.

[8] J. Määttä, A. Hadid and M. Pietikäinen, “Face spoofing detection from single images using micro-texture analysis”, 2011 International Joint Conference on Biometrics (IJCB), IEEE, 2011, pp. 1-7.

[9] A.R. Sadeghi, T. Schneider and I. Wehrenberg, “Efficient privacy-preserving face recognition”, International Conference on Information Security and Cryptology, Springer, Berlin, Heidelberg, 2009, pp. 229-244.

[10] C. Rathgeb and A. Uhl, “A survey on biometric cryptosystems and cancelable biometrics”, EURASIP Journal on Information Security, vol. 2011, no. 1, pp. 1-25, 2011.

[11] I. Chingovska, A. Anjos and S. Marcel, “On the effectiveness of local binary patterns in face anti-spoofing”,2012 BIOSIG-Proceedings of the International Conference of the Biometrics Special Interest Group (BIOSIG), IEEE, 2012, pp. 1-7.

[12] A.J. Aviv, K.L. Gibson, E. Mossop, M. Blaze and J.M. Smith, “Smudge attacks on smartphone touch screens”, Proceedings of the 4th USENIX conference on Offensive technologies (WOOT), Berkeley, CA, USA, USENIX Association, vol. 10, 2010, pp. 1-7.

Received: 31 May 2019/Accepted: 18 June 2019

[13] "Standards for Biometric Technologies", NIST, 2013. [Online]. Available: https://www.nist.gov/speech-testimony/standards-biometric-technologies. [Accessed: 01- Oct- 2018].
[14] D.F. Smith, A. Wiliem and B.C. Lovell, "Face recognition on consumer devices: Reflections on replay attacks", IEEE Transactions on Information Forensics and Security, vol. 10, no. 4, pp. 736-745, 2015.
[15] S. McQuiggan, L. Kosturko, J. McQuiggan and J. Sabourin, Mobile Learning, John Wiley & Sons, Canada, 2015.
[16] A. Buriro, S. Gupta and B. Crispo, "Evaluation of motion-based touch-typing biometrics for online banking", 2017 International Conference of the Biometrics Special Interest Group (BIOSIG), 2017, pp. 1-5.
[17] J. Marous, "Millennials Are Leading the Digital Banking Revolution", The Financial Brand. [Online]. Available: https://thefinancialbrand.com/64369/millennials-mobile-banking-digital-engagement-trends/. [Accessed: 01- Oct- 2018].
[18] A.S. Reid, "Financial crime in the twenty-first century: the rise of the virtual collar criminal", in White Collar Crime and Risk 2018, Palgrave Macmillan, London, 2018, pp. 231-251.
[19] N.C. Nguyen, O.J. Bosch, F.Y. Ong, J.S. Seah, A. Succu, T.V. Nguyen and K.E. Banson, "A systemic approach to understand smartphone usage in Singapore", Systems Research and Behavioral Science, vol. 33, no. 3, pp. 360-380, 2016.
[20] K. Aravindhan and R.R. Karthiga, "One-time Password: A Survey", International Journal of Emerging Trends in Engineering and Development, vol. 1, no. 3, pp. 613-623, 2013.
[21] N. Haller, C. Metz, P. Nesser and M. Straw, A one-time password system. RFC 1938, 1996.
[22] L. Lamport, "Password authentication with insecure communication", Communications of the ACM, vol. 24, no. 11, pp. 770-772, 1981.
[23] N. Haller, "The S/KEY one-time password system", RFC 1760, 1995.
[24] A.G. Chefranov, "One-time password authentication with infinite hash chains", in Novel Algorithms and Techniques In Telecommunications, Automation and Industrial Electronics, Springer Netherlands, 2008, pp. 283-286.
[25] K. Bicakci and N. Baykal, "Infinite length hash chains and their applications", in Enabling Technologies: Infrastructure for Collaborative Enterprises, WET ICE 2002, Eleventh IEEE International Workshops on, IEEE, 2002, pp. 57-61.
[26] R.L. Rivest, A. Shamir and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", Communications of the ACM, vol. 21, no. 2, pp. 120-126, 1978.
[27] SecureID, R. S. A. June 2003.
[28] T. Bradley, "RSA SecurID Hack Shows Danger of APTs", PCWorld, 2011. [Online]. Available: https://www.pcworld.com/article/222555/rsa_securid_hack_shows_danger_of_apts.html. [Accessed: 01- Oct- 2018].
[29] D. M'Raihi, M. Bellare, F. Hoornaert, D. Naccache and O. Ranen, "Hotp: A hmac-based one-time password algorithm", RFC4226, 2005.
[30] S. Hallsteinsen and I. Jorstad, "Using the mobile phone as a security token for unified authentication", in Systems and Networks Communications, ICSNC 2007, Second International Conference on, IEEE, 2007, pp. 68-68.
[31] Z. Zeltsan, S. Patel, I. Faynberg and A. Brusilovsky, "Password-Authenticated Key (PAK) Diffie-Hellman Exchange", RFC 5683, 2010.
[32] S. Liao, Q. Zhang, C. Chen and Y. Dai, "A unidirectional one-time password authentication scheme without counter desynchronization", in Computing, Communication, Control, and Management, CCCM 2009, ISECS International Colloquium on, vol. 4, IEEE, 2009, pp. 361-364.
[33] D. M'Raihi, S. Machani, M. Pei and J. Rydell, "Totp: Time-based one-time password algorithm", RFC6238, 2010.
[34] M.H. Eldefrawy, K. Alghathbar and M.K. Khan, "OTP-Based Two-Factor Authentication Using Mobile Phones", in Information Technology: New Generations (ITNG), 2011 Eighth International Conference on,. IEEE, 2011, pp. 327-331.
[35] W.B. Hsieh and J.S. Leu, "Design of a time and location based One-Time Password authentication scheme", in Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International, IEEE, pp. 201-206, 2011.
[36] C.L. Tsai, C.J. Chen and D.J. Zhuang, "Secure OTP and biometric verification scheme for mobile banking", in Mobile, Ubiquitous, and Intelligent Computing (MUSIC), 2012 Third FTRA International Conference on, IEEE, 2012, pp. 138-141.

Received: 31 May 2019/Accepted: 18 June 2019