ATLANTIS PRESS
Advances in Engineering Research, volume 118
2nd International Conference on Automation, Mechanical Control and Computational Engineering (AMCCE 2017)

An Enhanced SMS-based OTP Scheme

Yonghe Zhou[1, a], Liang Hu[1,b], Jianfeng Chu*[1,c]
[1]College of Computer Science and Technology, Jilin University,Changchun, 130012, China
[a]zhouyh14@mails.jlu.edu.cn, [b]hul@jlu.edu.cn, [c]chujf@jlu.edu.cn

Keywords: Short Message Service, One Time Password, Asymmetric Encryption

Abstract. Authenticate our identity by sending a One Time Password (OTP) through Short Message Service (SMS) is widely used in nowadays. But SMS-based OTP faces with some problems make it not secure. This paper presents a way by using encryption to enhance the security of SMS-based OTP. We introduced a parameter to make sure the request for SMS-based OTP is sponsored by the real user and then the OTP is send from the server to the user by encryption. It can prevent man in the middle (MITM), eavesdropping, replay attack and forgery attack.

Introduction

SMS plays an important role in our life. As mobile phone is a device that always carried, SMS is a good media to authenticate our identity. Now it is used by many internet services. We need to input a verification code when login websites which use two-factor authentication. The verification code is sent from the website server to our mobile phone by SMS to confirm "what we have" while static password is also needed to confirm "what we know". In addition, we need verification code to use E-bank service, reset password and so on. As one kind of OTP, verification code based on SMS has some security problems:

Communication Security. In GSM (2G), there are several flaws. First, mobile phone can't verify the authenticity of Base Station [1].Second, the ciphering algorithms are not strong enough to defend the outside attack. Although UMTS (3G) and LTE (4G) are stronger than GSM and they have gained widespread use in nowadays, but attackers can still attack mobile phone by shield UMTS and LTE signal.

Third-Party Service Provider Security. Not all the companies have their own device to send SMS. Sometimes these companies generate OTP and send it to mobile phone by the third-party service. At first, the OTP needs to send from these companies to the third-party service provider by internet. Attackers can eavesdrop the OTP if the communication is not safe enough or the third-party itself is not reliable enough.

Terminal Security. Mobile phone has experienced a process from feature phone to smartphone and also brings some weakness to SMS-based OTP. The capacity of feature phone are limited to some few functions and we can't install software on it. We only use it to receive verification code while the other operations are managed on computer when we use internet services or E-bank services. In this procedure, request and receive OTP are occurred in two independent devices that make it has a strong security performance. Smartphone are more powerful than feature phone. We can almost do everything on it just like a PC. These characters lead to all internet services and E-bank services can occurred just on our smartphone. Thus, request and receive OTP are operated in one device. On the other hand, applications on smartphone can get the permission to read SMS and connect to the Internet easily such as an Android smartphone. These applications can pretend to be a smartphone user to send OTP request to server and deal with it by Internet. This action against the confidentiality of information security and makes SMS-based OTP on smartphone is not safe enough as feature phones.

Related Work

Many solutions has been suggested to make SMS more secure. In a study by Agoyi and Seral [2], it compares the performance of three different asymmetric encryption techniques used on SMS: RSA cryptosystem, ELGamal cryptosystem and Elliptic curve cryptosystem. Kashif [3] introduced a way to secure SMS communication using encryption gateway and digital signature.

There are some other solutions to generate OTP by the smartphone but not receive it from the server. Eldefrawy and Alghathbar [4] introduced a way by add a challenge to prove the user's identity. The server send a random challenge to the user. After received the challenge, the user calculate his session OTP and send it to the server to authenticate himself. Aloul and Zahidi [5] propose a software token system to generate OTP. It uses IMEI number, IMSI number, username, PIN and current time as parameters to generate OTP.

Mulliner [6] summarize the SMS-based OTP thread model and proposed a way by using dedicated SMS-based OTP channel to secure it.

Most of these works regard smartphone as a completely secure device, ignore that other applications on smartphone can get the permission to access to SMS. Owning this, we propose an enhanced SMS-based OTP scheme. In this scheme, we consider the other applications are not trusted and they might have the permission to access to SMS information.

Our Approach

The key problem makes SMS-based OTP not secure is that others can get the communication content between the server and the end user. This problem break the confidentiality of information security. We can encrypt the SMS text to prevent others get the plaintext of it. Others can't understand it even if they can access to the ciphertext. As OTP is sent from the server to the end user, we need to keep the security of the entire procedure. If we just do some work between the server and the smartphone, it is still unsafe because the other applications on the smartphone can access to the SMS. Encryption are divided into symmetric encryption and asymmetric encryption. The both sides of a communication need to negotiate the key in symmetric encryption. But in asymmetric encryption they just need to know the public key of each other. While send verification code is a single way by the server to the user and the information is limit to a few characters, it's much easier to use asymmetric encryption than symmetric encryption. We introduced a seed parameter to ensure the OTP would not reveal to other applications on the smartphone.

Registration Protocol.

There are four steps in this section (Fig. 1).

1. User generate a random number that called S_{ini} which is only know by the user.
2. Smartphone initialize a counter called C_{cli} with the value 0.
3. Smartphone send S_{ini} and the user's other information to server by Internet.
4. Server store S_{ini} and user's other information, then initialize a counter called C_{ser} with the value 0.

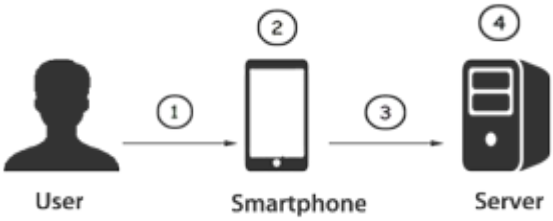

Fig. 1. Registration Protocol

Authentication Protocol.

There are five steps in this section (Fig. 2).

1. User input S_{ini} to the smartphone.

2. Smartphone generate a request code called U_{code} (Eq. 1) by HOTP [7] with the parameter S_{ini} and C_{cli}. A pair of public key K_{pub} and private key K_{pri} is also generated by asymmetric encryption algorithm.

$$U_{code} = HOTP(S_{ini}, C_{cli}) \qquad for\ C_{cli} = 1, 2, 3, ..., N \tag{1}$$

3. Server generate a request code called S_{code} (Eq. 2) with the same algorithm of step 2 and then compare it with U_{code}. If they are same, server will generate OTP called V_{otp} and encrypt it by K_{pub} to get E_{otp} (Eq. 3), then send E_{otp} to smartphone by SMS. If they are not same, the server will ignore this request.

$$S_{code} = HOTP(S_{ini}, C_{ser}) \qquad for\ C_{ser} = 1, 2, 3, ..., N \tag{2}$$

$$E_{otp} = K_{pub}(V_{otp}) \tag{3}$$

4. Smartphone get E_{otp} through SMS and decrypt it by K_{pri} to get V_{otp} (Eq. 4).

$$V_{otp} = K_{pro}(E_{otp}) \tag{4}$$

5. Smartphone authenticate it's identity by sending V_{otp} to server. If the authenticate is passed then C_{cli} and C_{ser} will plus one at the same time to make sure they are synchronous.

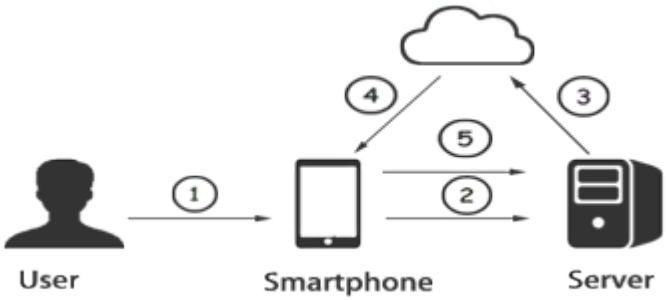

Fig. 2. Authentication Protocol.

Use HOTP algorithm guarantee others can't apply a replay attack and also make sure the user can't deny his request. Encryption make sure the SMS is confidential. These characters can prevent the tap from communication line and the malicious applications.

Security Analysis

In our proposal, the only thing that need to keep secret is S_{ini}. Other applications can't get the seed although they can get the ciphertext as they might have the permission to access to SMS.

MITM Attack. MITM attack contains two sides: the connection by Internet and the connection by SMS. The connection by Internet is through HTTPS, attackers can't modify the communication content. The connection by SMS is encrypted by K_{pub}, attackers also can't modify it as they don't know K_{pri}.

Eavesdropping. Attackers on the communication channel can eavesdrop the ciphertext which is encrypted by K_{pub}. While K_{pri} is generated by smartphone and is only known to itself, attackers can't decrypt ciphertext. Thus, they can't get the plaintext of OTP.

Replay Attack. In our approach, we generate U_{code} in smartphone and send it to server to check if it matches S_{code}. After a success procedure, C_{cli} and C_{ser} will plus one. As U_{code} is related to C_{cli}, it will be different in the next time. If attackers replay this request, U_{code} would not matches to S_{code}, server will ignore this request.

Forgery Attack. U_{code} is needed to prove the request is sponsored by the user. Attackers can't access to S_{ini} and they can't generate U_{code}. Thus, they can't implement forgery attack.

Conclusions

Today, we use many internet services, some of them are very important to us especially when it concerns our privacy and money. While enjoying these services, we also face with risks. In this paper, we proposed a scheme to enhance the security of SMS-based OTP. In this scheme, we take SMS as transport layer. Message are encrypted in this layer. Other applications can get the message of transport layer but they can't decrypt it. The decrypt work proceeded in application layer. This scheme can prevent outside threats like MITM attack and replay attack. It also can prevent threats from applications on smartphone like eavesdropping and forgery attack.

In the course of this work, we realize that the user might change his smartphone, the counter stored in smartphone might get lost. In our future work, TOTP [8] will be introduced to calculate the request code. Thus, the counter will be replaced by a time parameter which do not need to store on smartphone.

Acknowledgements

This work was financially supported by the European Seventh Framework Program (FP7) (GA-2011-295222) and the National Sci-Tech Support Plan of China (2014BAH02F03).

References

[1] Quirke J. Security in the GSM system[J]. AusMobile, May, 2004: 1-26.

[2] Agoyi M, Seral D. SMS security: an asymmetric encryption approach[C]//Wireless and Mobile Communications (ICWMC), 2010 6th International Conference on. IEEE, 2010: 448-452.

[3] Kashif M. Secure SMS Communication Using Encryption Gateway and Digital Signature[C]//Computational Science and Engineering (CSE), 2014 IEEE 17th International Conference on. IEEE, 2014: 1430-1434.

[4] Eldefrawy M H, Alghathbar K, Khan M K. OTP-based two-factor authentication using mobile phones[C]//Information Technology: New Generations (ITNG), 2011 Eighth International Conference on. IEEE, 2011: 327-331.

[5] Aloul F, Zahidi S, El-Hajj W. Multi factor authentication using mobile phones[J]. International Journal of Mathematics and Computer Science, 2009, 4(2): 65-80.

[6] Mulliner C, Borgaonkar R, Stewin P, et al. SMS-based one-time passwords: attacks and defense[C]//International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. Springer Berlin Heidelberg, 2013: 150-159.

[7] M'Raihi D, Bellare M, Hoornaert F, et al. Hotp: An hmac-based one-time password algorithm[R]. 2005.

[8] M'Raihi D, Machani S, Pei M, et al. Totp: Time-based one-time password algorithm[R]. 2011.