SMS-Based One-Time Passwords: Attacks and Defense
(Short Paper)

Collin Mulliner[1], Ravishankar Borgaonkar[2],
Patrick Stewin[2], and Jean-Pierre Seifert[2]

[1] Northeastern University
crm@ccs.neu.edu
[2] Technische Universität Berlin
{ravii,patrickx,jpseifert}@sec.t-labs.tu-berlin.de

Abstract. *SMS-based One-Time Passwords* (SMS OTP) were introduced to counter phishing and other attacks against Internet services such as online banking. Today, SMS OTPs are commonly used for authentication and authorization for many different applications. Recently, SMS OTPs have come under heavy attack, especially by smartphone Trojans. In this paper, we analyze the security architecture of SMS OTP systems and study attacks that pose a threat to Internet-based authentication and authorization services. We determined that the two foundations SMS OTP is built on, cellular networks and mobile handsets, were completely different at the time when SMS OTP was designed and introduced. Throughout this work, we show why SMS OTP systems cannot be considered secure anymore. Based on our findings, we propose mechanisms to secure SMS OTPs against common attacks and specifically against smartphone Trojans.

Keywords: Smartphone, OTP, SMS, mTAN, Malware, Multi-factor.

1 Introduction

Short Message Service (SMS) [1] based *One-Time Passwords* (OTP) were introduced to counter phishing and other attacks against authentication and authorization of Internet services. In these scenarios, SMS OTPs are mostly used as an additional factor in a multi-factor authentication system. Users are required to enter an OTP after logging in with a user name and password, or the OTP is required to authorize a transaction [8,21,24,13]. The prime example of SMS OTP is the *mobile Transaction Authorization Number* (mobile TAN or mTAN) that is used to authorize transactions for online banking services.

Unfortunately, today SMS OTP cannot be considered secure. Two reasons contribute to this fact. First, the security of SMS OTP relies on the confidentiality of SMS messages that in turn heavily relies on the security of cellular networks. Lately, several attacks against GSM and even 3G networks have shown that confidentiality for SMS messages cannot necessarily be provided. Second,

K. Rieck, P. Stewin, and J.-P. Seifert (Eds.): DIMVA 2013, LNCS 7967, pp. 150–159, 2013.

criminals have adjusted and created specialized mobile phone Trojans [3,17,9,15], since many service providers adapted SMS OTP to secure transactions.

To the best of our knowledge, so far nobody has studied the weaknesses of SMS OTPs in-depth, nor offered any solution that protects against specialized Trojans. In this work, we seek to improve the security of SMS-based one-time passwords. We investigate attacks against SMS-based one-time passwords in general and analyze attacks that are currently used in the real world. Through this analysis, we show that the perception of SMS messages as secure is probably false. In today's world, one would expect that OTPs are transported using end-to-end security. Our work shows that this is not true anymore. Our argument is based on facts and observations in two areas, cellular network infrastructure and the design of mobile phone as well as smartphone hardware and software.

Based on the results of our analysis, we investigate security enhancements for SMS OTPs. We design two solutions, and implement and evaluate the most promising one. Our primary solution, a virtual dedicated OTP channel, only requires minimal modification of the mobile phone operating system (OS) to secure SMS-based OTPs against common attacks. Our solution is completely backwards compatible since it does not require modification of the SMS or OTP message. The solution is implemented entirely as software modifications to the mobile phone. We created a demo video of our OTP channel solution running on a real Android phone: `http://www.youtube.com/watch?v=SF2HoK0D3%5F4`

Contributions. In this work we analyze the various attacks and weaknesses of SMS OTPs. We *identify the root causes for the insecurity of SMS OTP* today. The analysis provides the basis for the design of countermeasures. Our proposed defense mechanism, the *virtual dedicated channel*, protects against mobile phone Trojans and requires only a minor modification of the mobile phone operating system. Our solution is completely backwards compatible to currently deployed SMS OTP systems.

2 One-Time Passwords via SMS

One-Time Passwords. (OTP) are utilized as an additional factor in multi-factor authorization/authentication applications. They are only valid for exactly one authorization or authentication request. To avoid password lists, a convenient way to provide the user with an OTP is to send it via SMS. The phone number of the user must be registered for the service that provides SMS OTPs for authentication or authorization. OTPs are quite popular as an additional authorization or authentication factor in web-based services. These passwords can be utilized to *authenticate* a user, i. e., the user needs a valid OTP to prove his identity to log into a web application or to access the company's private network [8,21,26,24]. SMS OTPs are also used for account verification, e. g., Google Mail [13]. Recently, the online storage service Dropbox added SMS-based two factor authentication after facing some security issues. Online games such as Blizzard's Battle.net have also started using SMS for account unlocking. Another application for OTPs is

authorization. Here, the OTP is bound to a certain request or transaction in order to confirm it. Additionally, the OTP can be restricted to a very short time window. In online banking web applications for example, the user has to authenticate himself via a valid username and password to initiate a transaction. Directly after this transaction request, the user gets an SMS message containing the OTP that must be additionally entered to authorize the transaction. In this application area the OTP is called a *mobile Transaction Authorization Number* (mobile TAN or mTAN).

3 SMS OTP Threat Model

The attacker's goal is the acquisition of the OTP, and for this he has several options such as wireless interception or mobile phone Trojans. Less known attacks such as the SIM Swap Attack [14] can also be used. Below we further discuss the widely used attacks. Note that as the attacks target SMS interception in general, they can be used against all SMS OTP systems.

3.1 Wireless Interception

The GSM technology is insecure due to several vulnerabilities such as a lack of mutual authentication and weak encryption algorithms. Further research shows that the communication between mobile phones and base stations can be eavesdropped and decrypted using protocol weaknesses [4,5]. The attack framework presented by Nohl et al. can be used to intercept mobile traffic (GSM) of a dedicated end user, including SMS messages [20]. Lately, it has been shown that femtocells (small 3G base stations that are deployed in user homes) can be abused to intercept 3G communication, including SMS messages [11]. The attack works by installing a modified firmware on the femtocell that contains sniffing and interception capabilities. Furthermore, the report [19] suggests that such devices can be used to mount attacks against mobile devices by online criminals.

3.2 Mobile Phone Trojans

Mobile phone malware, and especially Trojans, that are designed to intercept SMS messages containing OTPs, are a rising threat. This kind of malware is created by criminals directly for the purpose of making money. In the following, we provide an overview of the different kinds of SMS OTP stealing Trojans.

The ZITMO (Zeus In The MObile) [3] Trojan for Symbian OS is the first known piece of malware that was specifically created for intercepting mTANs. The ZITMO binary is delivered as a normal signed Symbian application. It possesses the required capabilities in order to register itself with the Symbian OS to receive SMS messages when they arrive from the mobile network. Upon reception it can forward SMS messages to a predefined mobile number. Besides the capability to forward SMS messages, ZITMO can also delete SMS messages. This capability can be used to completely hide the fact that an SMS message

containing an mTAN ever arrived at the infected phone. Further, the ZITMO Trojan can be remotely reconfigured via SMS. Through this the attacker can, for example, change the destination number for forwarded SMS messages. In February 2011, a ZeuS version for Windows Mobile was detected and named Trojan-Spy.WinCE.Zbot.a [17]. The Trojan contained the same basic functionality as ZITMO. Similar Trojans also exist for Android [9] and RIM's Black Berry [10]. There are other Android Trojans that leverage access to SMS OTPs such as the MMarketPay.A [25] Trojan. This Trojan buys items from online stores and intercepts the SMS messages containing a verification code that is needed to complete the payment process. Additionally, further mobile malware, which steals authentication credentials, attacks mobile phone owners [22,27].

All known SMS OTP Trojans are user-installed malware. This means they do not leverage any security vulnerability of the affected platform. Instead, they use social engineering to trick the user into installing the binary. Further, the Trojans are executed as normal applications without special privileges.

4 Analysis of Weaknesses and Attacks

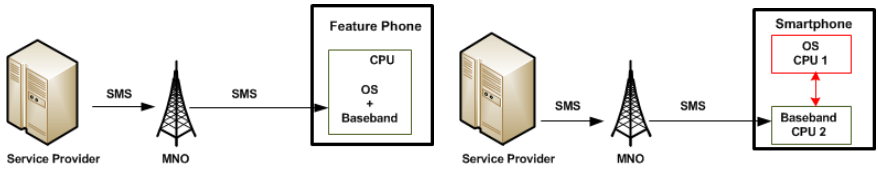
In this section, we analyze and discuss the security issues and attacks presented in Section 3. We identify and present the general reasons why certain weaknesses exist and why attacks are possible.

4.1 Cellular Network Insecurities

One major issue of SMS OTPs is that authentication service providers blindly rely on security provided by the mobile network operator (MNO). However as described in Section 3.1, numerous vulnerabilities in cellular network technologies suggest that it is possible to intercept cellular network traffic (in case of GSM). In addition, in some countries such as India, cellular network traffic is not encrypted by default. Furthermore, mobile network operators disable wireless encryption of SMS and call traffic. This can happen to decrease network load. Sometimes it occurs because of technical difficulties or because of a disaster such as an earthquake [7]. In these cases, an attacker equipped with suitable tools can intercept traffic to capture authentication codes transmitted over-the-air. However, one could argue that such personalized attacks against the authentication systems are less likely to happen and difficult to achieve in practice. Our goal is to stress that such new attacks prove that the fundamental assumption of considering cellular networks as a secure element and transmitting authentication codes in plain text cannot provide end-to-end security.

4.2 Mobile Phone Design Issues

Most mobile OSes provide an API to access received SMS messages from the SMS inbox. An OS can alternatively provide an API that allows an application to actively participate in the delivery process of SMS messages on the phone.



(a) The restricted OS of feature phones protects SMS messages.

(b) SMS messages are usually less protected once they left the separated baseband environment.

Fig. 1. Revealing End-to-End Security Deficiencies of Modern Smartphones

If the latter is possible, a Trojan can receive, alter, delete, and forward SMS messages without user interaction and without leaving a trace of its malicious behavior. By examining the hardware design of modern smartphones, we get a clearer picture of what has happened to the basic assumptions of the security of SMS messages. In the past mobile phones only consisted of one system, as shown in Figure 1(a), where one CPU executes both the mobile operating system and the baseband (the cellular interface). Smartphones today consist of two dedicated systems (two CPUs), as shown in Figure 1(b), one for the mobile operating system (e. g., Android) and one for the baseband. To protect the security-critical baseband, feature phone OSes were very restricted compared to smartphone OSes. This restriction helps to protect SMS messages on feature phones. Due to the described separation, baseband security is not the concern of the smartphone OS. As a result, smartphone OSes became very open. This means manufacturers are able to provide, among other things, very sophisticated APIs to the cellular subsystems such as SMS messaging. The main issue we identified is that SMS OTP was designed at a time where a mobile phone was a simple and dedicated system. This system was the endpoint for SMS messages. Legitimate applications could not access SMS messages on those phones, neither could Trojans. On smartphones, end-to-end security, as present on feature phones, does not exist anymore. Some smartphone OSes protect SMS messages through their permission system. Unfortunately, most users grant any permission to any application [23]. In Section 5.2, we present a protection mechanism to protect SMS messages while they are transported within the smartphone OS.

5 Defending SMS OTP

In this section, we present possible countermeasures that mitigate attacks against SMS OTP systems. We investigate approaches that require support of service providers, cellular network operators, and mobile OS manufacturers.

5.1 SMS End-to-End Encryption

Our first idea is to use end-to-end encryption to protect OTP messages when the SMS message gets intercepted or eavesdropped on. The idea relies on a concept

called *application private storage* that is found on almost all mobile platforms today. This is a permanent storage area that is private to each application. Only the application that stored a piece of data is able to access it. This kind of private storage is available on most of the common smartphone platforms such as Apple iOS, Google Android, Symbian OS, Windows Phone 7, and Java 2 Platform, Micro Edition (J2ME). The Android Data Storage description [12] states *"You can save files directly on the device's internal storage. By default, files saved to the internal storage are private to your application and other applications cannot access them (nor can the user). When the user uninstalls your application, these files are removed."* Windows Phone 7 and iOS have a similar model [18,2].

The concept is as follows. The OTP service generates the OTP message. For this it can keep its existing setup. In the second step the OTP message is encrypted with a customer-specific key. Each of the service's customers has a unique secret key. The encrypted OTP message is sent to the customer's mobile phone via SMS. This uses the existing OTP infrastructure operated by the service. On the user's phone, a dedicated application decrypts and displays the OTP message to the user. While an SMS OTP Trojan can still access the SMS message it cannot access the key that is required to decrypt the OTP message. The downside of this approach is the key distribution. Key distribution can be solved in many ways. We decided to not solve key distribution and rather investigate other solutions.

5.2 Virtual Dedicated Channel on the Handset

We identified mobile phone Trojans as the major threat to SMS OTP since the Trojan attack can be easily performed on a large scale. Hence, we present the following solution to protect against Trojan attacks that requires minimal support from operating system manufacturers and minimal-to-no support from the service provider and cellular network operators. Our solution is therefore very easy to deploy. Our main idea is to protect *certain* SMS messages against local interception by delivering them only to a specific application on the phone. Normally, any SMS capable application can read any SMS message that is received by the phone, as we discussed in Section 4.2. We create a *virtual dedicated channel inside the mobile phone OS* by removing *certain* SMS messages from the general delivery process on the phone and redirecting them to a special OTP application. Messages sent via this dedicated channel are secure against local interception. The endpoint of the virtual dedicated channel is an application with similar functionality to the default SMS application. It receives and stores SMS messages. The only difference is that it will only receive OTP messages, and that its message store cannot be read by other applications. The protection is ensured by the use of application private storage. From now on, we refer to this as the *OtpMessages* application. The *OtpMessage* application would be a pre-installed application that cannot be replaced in order to prevent Trojans form posing as the OTP application. Our dedicated channel is based on a minor modification of the mobile operating system. The modification is small since all mobile phones already implement specialized local routing of SMS messages to

implement the various features present in the SMS standard, e. g., WAP push. In Section 6, we will discuss the dedicated channel in detail.

6 Dedicated SMS OTP Channel

In the following, we present two design approaches. The first approach is based on SMS ports that represents a low effort and a clean design approach. The second approach is based on a message filter and offers backward compatibility and thus is easy to deploy. We implemented and evaluated the filter-based approach.

6.1 SMS Port-Based Channel

The SMS standard supports directing messages to specific applications via the use of SMS ports (similar to TCP/UDP ports) implemented using the *User Data Header* (UDH) [1]. The idea is to pick a port that is going to be used for OTP messages. The *OtpMessages* application will listen on this port to receive all OTP messages. To make sure that Trojans cannot bind to this port, operating system assistance is required. In particular, the OS only allows an application with a specific cryptographic signature to bind to this port. Almost all mobile operating systems support both required components: signed applications and SMS message routing based on ports. There are two minor challenges for this approach. First, the mobile operating system would need to be modified to add support for the SMS port-application signature combination. Second, the services that send SMS OTP messages need to know if a specific phone supports the dedicated OTP channel, since messages sent to an unused port are simply discarded. Due to these issues, we decided to explore a different path that we present in the next section.

6.2 Message Filter-Based Channel

We came up with the filter-based channel to provide a solution that only requires a small change in the mobile phone OS and neither involves the service provider nor the cellular operator. Furthermore, we want to keep the solution backwards compatible with phones that do not implement our protection mechanism. This is achieved through the fact that we do not require the SMS OTP messages to be changed. Our method acts as a filter inside the mobile operating system's SMS receiving code. Therefore, this solution can be easily added into the existing infrastructure present in the mobile phone OS. Our filter inspects every incoming SMS message to decide if the message has to be forwarded to the dedicated channel receiver, the *OtpMessages* application, or if the message is routed through the OS's default SMS path.

We developed two kinds of filters that can be used for our purpose: (i) The keyword-based filter is a filter that matches a keyword or a set of keywords against the message body or the start of a message. The keywords would be either hard coded into the SMS routing subsystem or configurable through an interface

that is not reachable through an API. (ii) The sender-based filter is a filter that matches against the originator address of an SMS message. It could also match against all short codes. Short codes refer to 4 to 6 digit phone numbers. Such codes are mostly used to interact with paid services.

Implementation. Our implementation extends the `dispatchPdus(..)` method in `SMSDispatcher.java` at `com/android/internal/telephony` of the Android 4.0 source. Our modification contains function named filter() that is used to inspect every incoming SMS message. If filter() determines that the message contains an OTP it changes the routing of that message to be delivered only to the *OtpMessages* application. For our implementation we used `OTP, mTAN, mobileTAN,` and `securetoken` for identifying OTP messages.

7 Evaluation

To evaluate our approach, we reconstructed the SMS sniffing Trojan scenario. We implemented a simple SMS sniffing Trojan by registering for `android.provider.Telephony.SMS_RECEIVED` events. This is the way SMS messages are received by any application, including malware [27]. Our Trojan grabs SMS messages as soon as they arrive and pops up a message box to show "SMS intercepted" as well as the message text, thus providing immediate feedback when the message has been intercepted. In a second step, we implemented the *OtpMessages* application. The application registers to receive incoming SMS messages using the same method as our Trojan. Every time *OtpMessages* receives an SMS message, it will display a pop-up containing the message and the string "OTP Message Received". This way, we can easily distinguish between our two applications. For the actual evaluation we crafted a number of SMS messages that contain OTPs. We sent the crafted messages from another mobile phone to our test device. All messages that contained any of the keywords were only received by the *OtpMessages* application. To verify that our Trojan still works, we sent a few messages to the phone that do not contain the filter string. Those messages were received by the Trojan.

8 Related Work

Koot [16] provides a simple risk analysis of mTAN security for iOS as well as Android smartphones. The work fails to provide an in-depth study of the root causes of mTAN insecurity. They do not aim to secure mTAN, but rather try to link the mobile phone to the computer used for online banking.

Several studies conducted on mobile malware [22,27] show that authentication credential stealing mobile malware exists in the wild. In this work, we present countermeasures that specifically protect against mobile malware that is built to intercept and exfiltrate authentication credentials sent via SMS.

A large scale study [6] evaluated authentication schemes in general using three main characteristics: usability, deployability, and security. Their security

characteristics basically attest SMS OTP with maximum points besides two issues. These issues are: not *Resilient-to-Internal-Observation* and not *Resilient-to-Theft*. Our virtual dedicated channel makes SMS OTPs *Resilient-to-Internal-Observation* and thus increases the security of SMS OTP significantly.

9 Conclusions

We presented the virtual dedicated channel, a solution that secures SMS-based OTPs against SMS stealing mobile phone Trojans. Our solution is completely backwards compatible and only requires minimal changes on the mobile phone side. Thus, our solution is easy to deploy since it leaves the infrastructure at the service provider and the OTP message format unchanged.

SMS-based OTP is one of the most user friendly multi-factor authentication mechanisms today that does not require an additional device. We believe our solution provides the means to secure SMS OTPs against attacks and thus helps to prevent online account theft and fraud.

Acknowledgements. This work was partially-supported by DARPA grant no: KK1243 (DarkDroid).

References

1. 3rd Generation Partnership Project: 3GPP TS 23.040 - Technical realization of the Short Message Service (SMS) (September 2004), http://www.3gpp.org/ftp/Specs/html-info/23040.html
2. Apple Inc.: IOS Developer Library: Cryptographic Services (July 2012), http://developer.apple.com/library/ios/#documentation/Security/Conceptual/Security_Overview/CryptographicServices/CryptographicServices.html#//apple_ref/doc/uid/TP30000976-CH3-SW6
3. Apvrille, A.: Zeus In The Mobile (Zitmo): Online Banking's Two Factor Authentication Defeated (September 2010), http://blog.fortinet.com/zeus-in-the-mobile-zitmo-online-bankings-two-factor-authentication-defeated/
4. Barkan, E., Biham, E.: Conditional estimators: An effective attack on A5/1. In: Preneel, B., Tavares, S. (eds.) SAC 2005. LNCS, vol. 3897, pp. 1–19. Springer, Heidelberg (2006)
5. Biryukov, A., Shamir, A., Wagner, D.: Real time cryptanalysis of A5/1 on a PC. In: Schneier, B. (ed.) FSE 2000. LNCS, vol. 1978, pp. 1–18. Springer, Heidelberg (2001)
6. Bonneau, J., Herley, C., von Oorschot, P.C., Stajano, F.: The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes. In: Proceedings of the IEEE Symposium on Security and Privacy (2012)
7. GSMK Cryptophone: Questions about the Interception of GSM Calls (2012), http://www.cryptophone.de/en/support/faq/questions-about-the-interception-of-gsm-calls/
8. Duo Security: Modern Two-Factor Authentication, http://duosecurity.com

9. F-Secure: Threat Description: Trojan:Android/Crusewind.A (2011), http://www.f-secure.com/v-descs/trojan_android_crusewind_a.shtml
10. Fisher, D.: Zeus Comes to the BlackBerry (August 2012), http://threatpost.com/en_us/blogs/zeus-comes-blackberry-080712
11. Gold, N., Redon, K., Borgaonkar, R.: Weaponizing femtocells: The effect of rogue devices on mobile telecommunication. In: Proceedings of the 19th Annual Network and Distributed System Security Symposium (NDSS) (February 2012)
12. Google Inc.: Data Storage — Android Developers, http://developer.android.com/guide/topics/data/data-storage.html#filesInternal
13. Google Inc.: Verifying your account via SMS or Voice Call, http://support.google.com/mail/bin/answer.py?hl=en&answer=114129
14. icici Bank: What is SIM-Swap fraud?, http://www.icicibank.com/online-safe-banking/simswap.html
15. Klein, A.: The Song Remains the Same: Man in the Mobile Attacks Single out Android (July 2012), http://www.trusteer.com/blog/song-remains-same-man-mobile-attacks-single-out-android
16. Koot, L.: Security of mobile TAN an smartphones. Master's thesis, Radboud University Nijmegen (February 2012)
17. Maslennikov, D.: ZeuS in the Mobile is back (February 2011), http://www.securelist.com/en/blog/11169/Zeus_in_the_Mobile_is_back
18. Microsoft Coperation: Windows Phone 7 Security Model (December 2010), http://download.microsoft.com/download/9/3/5/93565816-AD4E-4448-B49B-457D07ABB991/WindowsPhone7SecurityModel_FINAL_122010.pdf
19. Muttik, I.: Securing Mobile Devices:Present and Future (December 2011), http://www.mcafee.com/us/resources/reports/rp-securing-mobile-devices.pdf
20. Nohl, K., Pudget, C.: GSM: SRSLY? (2009), http://events.ccc.de/congress/2009/Fahrplan/events/3654.en.html
21. PhoneFactor, Inc.: Comparing PhoneFactor to Other SMS Authentication Solutions, http://www.phonefactor.com/sms-authentication
22. Felt, A.P., Finifter, M., Chin, E., Hanna, S., Wagner, D.: A Survey of Mobile Malware in the Wild. In: Proceedings of the ACM Workshop on Security and Privacy in Mobile Devices, SPSM (2011)
23. Felt, A.P., Greenwood, K., Wagner, D.: The Effectiveness of Application Permissions. In: USENIX Conference on Web Application Development (2011)
24. SMS PASSCODE A/S: Two-factor Authentication, http://www.smspasscode.com/twofactorauthentication
25. TrustGo Mobile Inc.: MMarketPay.A (2012), http://blog.trustgo.com/mmarketpay-a-new-android-malware-found-in-the-wild-2/
26. VISUALtron Software Corporation. 2-Factor Authentication - What is MobileKey?, http://www.visualtron.com/products_mobilekey.html
27. Zhou, Y., Jiang, X.: Dissecting Android Malware: Characterization and Evolution. In: 33rd IEEE Symposium on Security and Privacy (May 2012)