Website Credential Storage and Two-Factor Web Authentication with a Java SIM

Jonathan Hart, Konstantinos Markantonakis, Keith Mayes

Website Credential Storage and Two-factor Web Authentication with a Java SIM

J. Hart, K. Markantonakis, K. Mayes

Information Security Group, Smart Card Centre
Royal Holloway University of London, Surrey, United Kingdom
{jon.hart, k.markantonakis, k.mayes}@rhul.ac.uk

Abstract. In this paper two mobile website authentication schemes are proposed. The first enables authentication credentials (username and password) to be stored and retrieved securely from a mobile handset, and requires no changes to existing websites. The second scheme, which may optionally be used with the first, utilises a one-time password and is intended for applications requiring an enhanced level of authentication, e.g. financial services. Both authentication schemes use a Java SIM and ubiquitous mobile phone; with its familiar and convenient form factor and high user acceptance. Both schemes also provide protection against online phishing attacks.

Key words: web authentication, credential storage, Java SIM, SIM toolkit, two-factor, passwords

1 Introduction

In recent years there has been an exponential growth in the number of e-commerce services available, leading to a proliferation in the number of credentials that a user is expected to remember to access these services. Adams and Sasse [1] report that the average user can typically only use four or five unrelated passwords effectively, and this number falls if passwords are used infrequently. Sophos [2] in a recent survey found that a third of respondents admitted to reusing the same password on multiple websites.

Users that reuse passwords on multiple sites may not realise that the security of a well protected account is only as good as that of the poorly-protected account. Blake Ives et. al [3] describe this as the domino effect. It is conceivable that a criminal could deliberately setup a malicious server specifically for the purpose of harvesting users' credentials in an attempt to re-use those credentials on other websites.

Financial institutions typically employ two-factor authentication to protect against key logging and phishing attacks, requiring a user to enter something they know (such as a username and password) and also something in their possession (such as a One-Time Password (OTP) from a hardware token or card reader). Hardware tokens and readers may however be inconvenient, especially if the user

has several accounts with different institutions; each requiring different hardware tokens or card readers.

Two main issues with existing authentication systems have been identified:

1. the proliferation of passwords required for authentication to an ever increasing number of internet sites; and
2. the inconvenience of incompatible two-factor authentication tokens and card readers.

In this paper we propose a convenient and secure solution to the above issues, using a mobile phone and Java SIM. The mobile phone is a familiar device which many people have within arms reach; this makes it attractive to use for authentication as the user is not burdened with additional hardware to carry around or learn to use. The loss or theft of a mobile phone is also likely to be reported more promptly than an authentication token. The SIM card is also a tamper resistant device which makes it an ideal candidate for the storage of authentication credentials and encryption keys. The SIM may also be transferred easily between handsets as required.

Several existing authentication systems commonly in use today were reviewed, including password managers and two-factor OTP solutions. A number of issues with these systems were identified and the motivation set for developing the two authentication systems described in this paper. Due to page limits these are not discussed any further, however for an extensive discussion please see [4].

Related work includes the protocol described by Wu et. al [5]; in which no change is required to the web server; however credentials are stored on a potentially vulnerable third party proxy server. Gouda et. al [6] describe a method for using a single password with multiple websites, without each site having knowledge of the original password, and Mannan and Oorschot [7] propose a scheme for strengthening password authentication using a personal device. Both of these schemes however require significant changes to server infrastructure. Florncio et. al [8] propose a protocol that allows users OTP access to any web account without requiring any changes to the server. Here a user is required to trust a reverse-proxy server during the registration phase as the OTPs are based on their secret credentials. Management of multiple accounts may also be an issue with multiple sets of OTPs to manage.

2 The Proposed Authentication Schemes

The proposed authentication schemes assume that a user's mobile number is associated with only one SIM card. The following design goals were considered:

- Provide a scheme for storing web credentials on a SIM.
- Provide a second more secure scheme for authenticating a user's unique SIM.
- Both schemes should be easy to operate and require either minimal or no changes to existing systems.
- The design should be portable so that it can easily used from any PC, including for example in an Internet Café, without requiring any special hardware.

2.1 The Entities Involved

Each of the entities in the system is now described, with reference to Fig. 1.

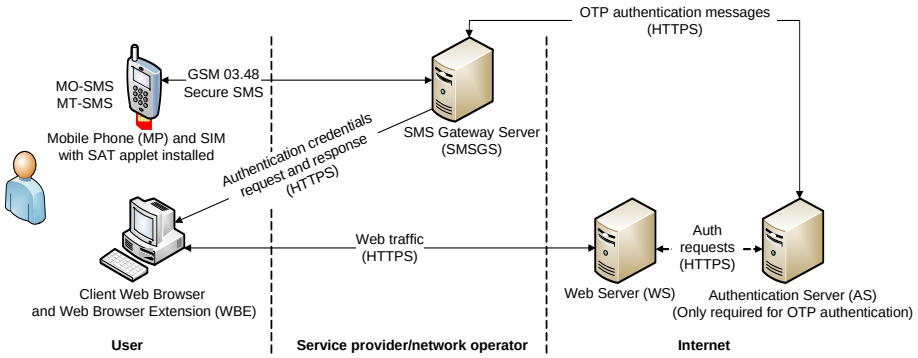

Fig. 1. System Architecture

SIM Application Toolkit (SAT) Applet. We have developed a Java SIM SAT [9–11] applet which executes on the users SIM card. The applet provides the credential storage and authentication services for the two authentication schemes proposed.

Mobile Phone (MP). The MP consists of a SIM toolkit capable handset with SMS capability and a SIM pre-loaded with the SAT applet. No software is installed on the mobile phone.

Web Browser Extension (WBE). A WBE integrates into the web browser environment to extend the functionality of the browser [4]. The WBE is a required entity in the first authentication scheme. The main function of the WBE is to enable the credentials stored by the SAT applet to be retrieved from the SIM and entered into a web page logon form with minimal user interaction.

SMS Gateway Server (SMSGS). The SMSGS role is to translate mobile network SMS messages to and from the SAT applet to HTTPS commands that are sent and received over a TCP/IP based network to the WBE and AS entities. SMS messages are sent using the GSM 03.48 secure SMS format [12, 13].

Authentication Server (AS). In the two-factor OTP based authentication scheme third party institutions are required to operate an AS. The AS shares an institution generated subscriber specific secret key with each subscribers SAT applet. The institution AS and SAT applet are the only two entities that share this secret.

Web Server (WS). The WS serves a user's chosen content in response to a request from a user's web browser, typically this will be over HTTPS for a secure website. In the case of the two-factor authentication scheme, the WS has a back-end connection to the AS which provides two-factor authentication for specific web pages or transactions.

2.2 Protocol Details.

The proposed authentication schemes are now discussed with reference to the notation summarised in Table 1.

Table 1. Notations, Keys and Algorithms used in protocol descriptions

Notation	Description
$Y \| Z$	Concatenation of the data items Y and Z in that order
$X \rightarrow Y : C$	The entity X sends the entity Y a message with contents C
$X \rightarrow Y \rightarrow Z : C$	The entity X sends the entity Z a message with contents C, via entity Y
$h(Z)$	Is the result of a collision resistant hash function such as SHA-2 applied to data string Z
$E_K(Z)$	Is the result of encryption of the data string Z with a symmetric algorithm such as AES or 3DES using key K
$PK_X(Z)$	Is the result of encryption of data string Z with a public key algorithm such as RSA and key X
$MAC_X(Z)$	Is the result of a keyed MAC of the string Z, with key X
X-RAND	Is a random number generated by entity X
X-PUB	Represents entity's X public encryption key
X-PRIV	Represents entity's X private encryption key
X-SYM	Represents entity's X symmetric encryption key
X-SEED	Represents a seed value for entity's X one way hash function
X-IDENTITY	A string representing entity's X identity
AUTH	Is the users authentication key
CRED-UID	The web authentication credential username
CRED-PWD	The web authentication credential password

Authentication Scheme 1 - Web Credential storage on SIM. In the first authentication scheme a user's website credentials are stored on the user's SIM card. The credentials are requested on-demand when the user visits a website requiring authentication.

1. The protocol starts with a one off-initialisation step that sets up authentication and encryption keys for the remainder of the protocol. The user installs the WBE on a host PC and chooses a username U and passphrase P, from which an authentication hash $AUTH$ and symmetric key *WBE-SYM* are

created in volatile memory:

$$WBE\text{-}SYM = h(P\|U) \text{ and } AUTH = h(WBE\text{-}SYM)$$

The above hashes are created using a collision resistant hash function such as SHA 256, in line with Ecrypt [14] recommend key lengths. *WBE-SYM* is the symmetric key used by the SAT applet to send credentials to the WBE. $AUTH$ is used to identify the user to the SMSGS. A message is then sent to the SMSGS to associated the username U with mobile number N in the SMSGS internal database:

$$\text{WBE} \rightarrow \text{SMSGS: } (U||AUTH||N) \tag{1.1}$$

In response to the receipt of message (1.1) the SMSGS stores a permanent user identity record in its database and sends a message to the SAT applet identified by mobile number N requesting the applet sends the WBE its RSA public key:

$$\text{SMSGS} \rightarrow \text{SAT: } (PublicKeyRequest) \tag{1.2}$$

The SAT applet responds to message (1.2) with the applets RSA public key *SAT-PUB* and forwards this via the SMSGS to the WBE:

$$\text{SAT} \rightarrow \text{SMSGS} \rightarrow \text{WBE: } (SAT\text{-}PUB) \tag{1.3}$$

The WBE responds with the *WBE-SYM* symmetric key, enciphered under the SAT applets RSA key, via the SMSGS gateway:

$$\text{WBE} \rightarrow \text{SMSGS} \rightarrow \text{SAT: } \text{PK}_{SAT\text{-}PUB}(WBE\text{-}SYM) \tag{1.4}$$

Message (1.4) completes the initialisation phase. To recap, at the end of this phase a user account has been created on the SMSGS and has been associated with the user's mobile number. A symmetric key has also been exchanged between the WBE and SAT applet. The initialisation phase is only carried out when the WBE passphrase is initially set or changed.

2. Following initialisation the WBE may request a user's credentials from the SAT applet, by supplying the username U, authentication hash $AUTH$ (derived from passphrase P; as before) and the website hostname S to the SMSGS:

$$\text{WBE} \rightarrow \text{SMSGS: } (U\|AUTH\|S) \tag{1.5}$$

$$\text{SMSGS} \rightarrow \text{SAT: } (GetCredential\|S) \tag{1.6}$$

On receipt of message (1.6) the SAT applet checks the internal credential store and following confirmation by the user entering their PIN, responds with the user's website credential username *CRED_UID* and password *CRED_PWD*. If no credentials are currently stored the user is given the opportunity to enter them. The credentials are then enciphered with the previously exchanged symmetric key *WBE-SYM* and sent to the WBE:

$$\text{SAT} \rightarrow \text{SMSGS} \rightarrow \text{WBE: } \text{E}_{WBE\text{-}SYM}(CRED\_UID\|CRED\_PWD) \tag{1.7}$$

On receipt of message (1.7) the WBE deciphers the credentials received using the pre-shared symmetric key, and enters them into the web page logon form. This completes the protocol run.

Authentication Scheme 2 - Two-factor OTP authentication. For the two-factor authentication scheme a OTP is generated by the SAT in response to a random challenge from the AS.

1. The protocol starts with a one off-initialisation step that sends the AS OTP seeding key to the SAT applet. The seeding key is enciphered under the SAT applets public RSA key, which is obtained using messages (1.1 - 1.3) above, substituting the AS for the WBE entity (not shown), the seeding key *AS-SEED* is then sent to the SAT applet along with the AS's identity *AS-IDENTITY*:

$$\text{AS} \rightarrow \text{SMSGS} \rightarrow \text{SAT: } \text{PK}_{SAT\text{-}PUB}(AS\text{-}IDENTITY \| AS\text{-}SEED) \quad (2.1)$$

   The SAT applet stores the AS's identity *AS-IDENTITY* and seeding key *AS-SEED*, after confirming this action with the user.
2. Following the one off-initialisation above, the AS may then authenticate a user and SIM when requested by the WS (for example to login to a web page, or approve a transaction). The authentication request message from the AS to the SAT takes the following form:

$$\text{AS} \rightarrow \text{SMSGS} \rightarrow \text{SAT: } \text{PK}_{SAT\text{-}PUB}(AS\text{-}IDENTITY \| AS\text{-}RAND) \quad (2.2)$$

   Where *AS-RAND* is a random number from a good and unpredictable source of randomness.
3. The SAT applet decrypts message (2.2) using its private RSA key and if a seed value has been previously stored for the AS with identity *AS-IDENTITY* then asks the user to enter their PIN to confirm the authentication. If confirmed the SAT applet generates a response to the challenge *AS-RAND*, using a keyed MAC function:

$$\text{SAT} \rightarrow \text{SMSGS} \rightarrow \text{AS: } \text{MAC}_{AS\text{-}SEED}(AS\text{-}RAND) \quad (2.3)$$

   The AS receives message (2.3) and compares it with the locally generated expected response. If the SAT applet response matches the expected response the user and SIM are authenticated, otherwise authentication fails. This marks the end of the protocol run.

3 Practical Implementation

To test the proposed authentication schemes we carried out a proof of concept practical implementation. This involved developing a SAT applet, SMSGS and WBE. The SAT applet was loaded onto a test card and real-time tests performed against live websites for the SIM credential storage scheme. The two-factor OTP based authentication scheme was validated using the SAT development environment simulator [15] due to restrictions on the test card crypto functionality. The practical implementation is discussed in detail in [4].

4 Security Analysis

RSA public key cryptography is used for both the authentication schemes to exchange symmetric keys that are used later on in the protocol; the security of these keys is therefore reliant on a sufficiently strong RSA key length and secrecy of the RSA private key. The RSA private key is generated internally within the SIM and no external method is provided to retrieve the private key. Key lengths should be chosen to provide an adequate level of protection as recommended by Ecrypt [14]. Public Key Infrastructure (PKI) is not implemented to verify the authenticity of the public RSA keys from the SIM. As the key pair is generated by the SIM and tied to the host SIM mobile number we felt that this provided a reasonable level of assurance within this closed system.

Web authentication credentials are encrypted before they leave the SIM applet with the symmetric key of the destination entity. This provides end-end encryption and protection of the credentials should the SMSGS be compromised. To guard against eavesdropping, man-in-the-middle and relay attacks, GSM 03.48 is used for secure SMS transport and the HTTPS between WBE, AS and SMSGS. HTTPS protects the integrity of the messages and also provides assurance that the SMSGS is genuine from the WBE side as the SMSGS SSL certificate is validated against a third party CA certificate during the initial HTTPS handshake.

As the WBE does not operate in a secure environment it may be possible for another web browser extension or screen scraper to get access to sensitive information (such as the login credentials or WBE passphrase), however the same risks apply when entering the credentials by hand into the browser.

In the OTP authentication design each AS shares a user specific individual secret key with the SIM only. As a result, even if an AS in the system were compromised only the keys owned by that AS would be vulnerable.

Both authentication methods implement a PIN lockout function (after three incorrect PIN attempts) in the SAT applet to prevent an exhaustive PIN search should the handset be lost or stolen.

5 Conclusions and Future Work

In this paper two authentication schemes using a Java SIM have been demonstrated. The first scheme does not require modifications to existing websites and reduces the risk of password re-use as a user no longer needs to remember their credentials for individual websites and can choose a more secure or randomly generated password. In fact the SAT applet functionality could be extended to generate random passwords, removing the risk of a user choosing a poor password. The authentication scheme also provides a degree of protection from phishing attacks; as a user inadvertently visiting a phishing site will not have their credentials stored for that site on the SIM. It is acknowledged that as the scheme uses existing form based password authentication this may not provide the level of security required by certain institutions.

The second authentication scheme provides an enhanced level of authentication as the user's SIM is authenticated in real-time using a random challenge generated by the authenticating parties AS. A single SIM could therefore replace the multiple and often incompatible OTP tokens and card readers.

A challenge to a wider scale implementation of the proposed schemes is the network operator's ownership and control of the SIM. A possible solution to this would be to implement a hybrid solution, using a lightweight SAT applet to provide credential storage (provided and installed by the network operator) and a MIDP Java applet that executes on the mobile handset using JSR177 API extensions to interface to the lightweight SAT applet. Care would however need to be exercised to ensure that code and data stored on the mobile handset was not vulnerable to attack.

References

1. Adams, A., Sasse, M.A.: Users are not the enemy. Commun. ACM **42**(12) (1999) 40–46
2. Sophos: at risk as one third of surfers admit they use the same password for all websites. `http://www.sophos.com/pressoffice/news/articles/2009/03/password-security.html`.
3. Ives, B., Walsh, K.R., Schneider, H.: The domino effect of password reuse. Commun. ACM **47**(4) (2004) 75–78
4. Hart, J., Markantonakis, K., Mayes, K.: Website credential storage and two factor web authentication with a java sim. Cryptology ePrint Archive, Report 2010 (2010)
5. Wu, M., Garfinkel, S., Miller, R.: Secure web authentication with mobile phones. In: DIMACS Workshop on Usable Privacy and Security Software. (2004)
6. Gouda, M.G., Liu, A.X., Leung, L.M., Alam, M.A.: Spp: An anti-phishing single password protocol. Comput. Netw. **51**(13) (2007) 3715–3726
7. Mannan, M., van Oorschot, P.C.: Using a personal device to strengthen password authentication from an untrusted computer. In: Financial Cryptography. (2007) 88–103
8. Florêncio, D., Herley, C.: One-time password access to any server without changing the server. In: ISC '08: Proceedings of the 11th international conference on Information Security, Berlin, Heidelberg, Springer-Verlag (2008) 401–420
9. 3GPP: Specification of the SIM Application Toolkit (SAT) for the Subscriber Identity Module - Mobile Equipment (SIM-ME) interface. TS 11.14, 3rd Generation Partnership Project (3GPP) (2007)
10. 3GPP: Subscriber Identity Module Application Programming Interface (SIM API) for Java Card. TS 03.19, 3rd Generation Partnership Project (3GPP) (2002)
11. 3GPP: (U)SIM Application Programming Interface (API); (U)SIM API for Java Card. TS 31.130, 3rd Generation Partnership Project (3GPP) (2009)
12. Guthery, S.B., Cronin, M.: Mobile Application Development with SMS and the Sim Toolkit. McGraw-Hill Professional (2001)
13. 3GPP: Security mechanisms for SIM application toolkit; Stage 2. TS 03.48, 3rd Generation Partnership Project (3GPP) (2005)
14. Ecrypt II: Report on key sizes. `http://www.keylength.com/en/3/`.
15. Gemalto NV: Gemalto Developer Suite. `http://www.gemalto.com`.