Security of the Lin-Lai smart card based user authentication scheme

Chris J. Mitchell and Qiang Tang

Technical Report
RHUL–MA–2005–1
27 January 2005

Department of Mathematics
Royal Holloway, University of London
Egham, Surrey TW20 0EX, England
http://www.rhul.ac.uk/mathematics/techreports

Abstract

The remote user authentication scheme of Lin and Lai, that uses a smart card and a fingerprint measurement, is reviewed and shown to possess significant security issues.

1 Introduction

Lin and Lai [3] propose a method for remote user authentication to a server using a combination of a password, a biometric, a smart card, and a protocol derived from the El Gamal signature scheme [1]. The protocol is a modified version of a similar scheme due to Lee, Ryu and Yoo [2].

Lin and Lai claim in [3] that the scheme is secure. However, no rigorous evidence of security is provided. We show in this paper that, in fact, the scheme has significant weaknesses that cast doubt on its useability.

2 The user authentication scheme

We first briefly review the operation of the scheme. The scheme has two main phases — user registration, and operational use of the scheme for user authentication (login). There is also a means defined to change a user password. All three of these phases are outlined below.

The scheme described in [3] is based on a specific biometric, namely fingerprint recognition. The smart card given to each user as a result of registration is equipped with a 'template' for the user fingerprint. However it is not clear whether or not this card is assumed to possess a fingerprint reader.

In the descriptions below we use h to denote a cryptographic hash-function (that inputs a string of bits of arbitrary length and outputs a fixed length string of bits), and $\oplus$ to denote the bit-wise exclusive-or operation on strings of bits (where, if the two bit strings are of unequal length, the shorter one is padded with zeros).

Prior to any user registrations, the server (to which the users will authenticate) must choose certain system parameters. In particular a prime P must be chosen and a secret exponent x must be chosen and held securely.

2.1 Registration

We suppose user A, with identifier i_A, is to be registered. It is stated that user identifiers must have a fixed format, although the nature of this format (and the ‘quantity’ of redundancy in this format) is not specified. It is also assumed that i_A can be treated as an integer.

Registration involves the following steps.

1. User A has his/her biometric template computed — for fingerprint recognition this will contain a specification of the fingerprint minutiae, denoted by s_A. For the purposes of the scheme we assume that this is a string of bits.

2. User A chooses a password, p_A. For the purposes of the scheme we assume that this consists of a string of bits.

3. The registration centre operated by the authentication server computes

$$y_A = ((i_A)^x \bmod p) \oplus h(p_A \oplus s_A)$$

   where the integer $(i_A)^x \bmod p$ is converted to a string of bits by some means (e.g. by taking the binary representation).

4. The registration centre stores i_A, s_A and y_A, together with the system parameter P, on a smart card, which is issued to user A. The smart card is assumed to be equipped with an implementation of the hash-function h.

2.2 Login

When user A is to be authenticated, the following steps are performed.

1. The user inserts the issued smart card into a card reader (remote to the authentication server) and also provides a fingerprint imprint. As mentioned above, it is not stated in [3] whether it is assumed that the fingerprint reader is integral to the smart card (e.g. as in [6, 7]), or built into the card reader. However, this makes a significant difference to the overall security of the scheme, as we discuss below. The user is further required to input a password. The entered password (p_A^*) and the fingerprint data (if gathered externally to the card) are passed to the card.

2. The card compares the captured fingerprint data with the stored biometric template s_A. If they do not match then processing is aborted.

3. The card generates a random integer r using the captured fingerprint data.

4. The card computes
$$C_1 = (i_A)^r \bmod P$$
and
$$C_2 = (y_A \oplus h(p_A^* \oplus s_A))^r.(h(y_A \oplus T \oplus h(p_A^* \oplus s_A))) \bmod P$$
where T is a newly generated timestamp, the integer y_A is converted to a string of bits, and where the strings of bits $y_A \oplus h(p_A^* \oplus s_A)$ and $h(y_A \oplus T \oplus h(p_A^* \oplus s_A))$ are converted to integers.

5. The card sends $C = (i_A, C_1, C_2, T)$ to the authentication server.

6. The authentication server checks the format of i_A for correctness (by some unspecified means).

7. The authentication server checks that the timestamp T is within the current ‘acceptance window’.

8. The authentication server computes
$$C_2(C_1)^{-x} \bmod P$$
and
$$h(((i_A)^x \bmod P) \oplus T) \bmod P$$
and accepts A only if the two agree (where $-x$ is computed modulo $P-1$).

2.3 Password change

When user A wishes to change his/her password p_A, the following steps are performed.

1. The user inserts the issued smart card into a card reader (remote to the authentication server) and also provides a fingerprint imprint.

2. The user inputs both the existing password p_A and the new password p_A^*.

3. The card computes
$$y_A^* = y_A \oplus (h(p_A \oplus s_A) \bmod P) \oplus (h(p_A^* \oplus s_A) \bmod P).$$

4. The card replaces y_A with y_A^*.

3 Analysis

We now describe a number of security issues with the Lin-Lai scheme. Some of these issues arise because of the lack of clarity in the description of the scheme, although others apply regardless of how the description is interpreted.

First observe that Lin and Lai [3] do not provide a threat model for their scheme, and hence it is not possible to be certain about what types of threat their scheme is designed to address. However, given that the scheme is designed as a remote user authentication scheme, we assume that the card reader (and attached device, e.g. a PC) into which the user card is inserted, are not necessarily trusted. As a result we consider the case below where a malicious third party has obtained a password for a stolen card, e.g. by persuading the legitimate user to perform an authentication process using a fraudulent or manipulated card reader. This corresponds to a claim in Section 1 of [3], where it is stated that the scheme is designed to withstand attacks where the password and card have been stolen.

It is also not stated in [3] where the timestamp T is derived from. We therefore assume that this is generated externally to the card, i.e. by the card reader or attached terminal, since smart cards are typically not able to maintain a clock. This is because such cards do not normally have a power supply.

- Suppose a malicious party has stolen a card and also knows the password for this card. If the malicious party can also arrange for a valid fingerprint sample to be provided to the card, then it is simple to see that the card can be misused to impersonate the legitimate user. How difficult providing a valid fingerprint sample to the card will be depends on whether the fingerprint reader is integral to the card. If it is not, then replaying a copied fingerprint sample to the card is trivially simple. If the reader is integral to the card then research of Matsumoto et al. [4, 5] suggests that, given a copy of the fingerprint of the legitimate cardholder, e.g. as left on a glass, then a false 'finger' can be manufactured from gelatine that will provide a valid reading. Thus, regardless of the location of the fingerprint reader, the system would appear vulnerable to loss of card and password, contrary to the claim made in [3].

- Since the protocol only has a single message exchange (from card to server), the authentication server relies entirely on the time stamp T to determine the 'liveness' of the card. However, as we have mentioned above, the timestamp will not be generated by the card, and the card cannot know whether or not a timestamp provided by a terminal is valid or not. As a result a malicious terminal could provide a timestamp T for some future time to a genuine card, which will generate a valid authenticator for this future time. The malicious terminal can then masquerade as the valid user at this future date.

- The system relies on valid user identifiers adhering to a particular format. However, this format is not specified in [3], and neither is any indication given of the properties that this format should possess. We now demonstrate the importance of this format, and indicate what rules should be used to select such a format.

  Suppose a malicious party E is able to observe two separate valid authentications conducted by the same card, i.e. to observe two separate messages sent to the authentication server. Suppose moreover that these two authentication messages both use the same timestamp T — this could be achieved by using a manipulated terminal, given that we are assuming that the timestamp is generated externally to the card. Suppose that the intercepted messages are (i_A, C_1, C_2, T) and (i_A, C_1', C_2', T), where the messages are computed using random values r and r' respectively. Then we immediately have:

  $$C_1.(C_1')^{-1} \bmod P = (i_A)^r.(i_A)^{-r'} \bmod P = (i_A)^{r-r'} \bmod P$$

  and

  $$\begin{aligned} C_2.(C_2')^{-1} \bmod P &= (y_A \oplus h(p_A \oplus s_A))^r.(h(y_A \oplus T \oplus h(p_A \oplus s_A))). \\ &\quad\ (y_A \oplus h(p_A \oplus s_A))^{-r'}. \\ &\quad\ (h(y_A \oplus T \oplus h(p_A \oplus s_A)))^{-1} \bmod P \\ &= (y_A \oplus h(p_A \oplus s_A))^{r-r'} \bmod P \\ &= (i_A)^{x(r-r')} \bmod P. \end{aligned}$$

  That is, E can immediately compute a pair of values $((i_A)^w \bmod P$, $(i_A)^{wx} \bmod P)$, for some (unknown) value of w.

  The interceptor E next computes a series of values $((i_A)^w)^j \bmod P$ for a range of integers j, until a value is found, $((i_A)^w)^k \bmod P$ say, which has the correct format for a user identifier. E now has the means to conduct a successful authentication for a (non-existent) user B with identifier $i_B = (i_A)^{wk} \bmod P$. This is achieved as follows.

  E first computes $z_B = ((i_A)^{wx})^k \bmod P = (i_B)^x \bmod P$. When E is required to impersonate B at time T, E chooses a random value r, computes

$$C_1 = (i_B)^r \bmod P$$

and

$$C_2 = (z_B)^r.(h(z_B \oplus T)) \bmod P,$$

and sends the message (i_B, C_1, C_2, T). It is simple to verify that this message will be accepted as genuine by the authentication server.

To avoid such an attack it is necessary to choose a format for identifiers that contains sufficient redundancy so that it will be computationally infeasible for an attacker to find a valid identifier by computing successive powers of a random integer.

Note that above analysis also serves to illustrate the fundamental idea behind the system, namely that the value $(i_A)^x \bmod P$ is the smart card's critical secret value. Compromise of a pair of values (i_A, $(i_A)^x \bmod P$), where i_A is a valid identifier, will enable unlimited masquerades as an entity with identifier i_A.

- As part of the motivation for the design of their scheme, Lin and Lai [3] describe an attack on a previously proposed, and somewhat similar, scheme due to Lee, Ryu and Yoo [2]. However, this attack assumes that if i_A is an identifier for the valid user A, then the identifier $(i_A)^t \bmod P$ will be accepted as valid, where t is a random integer. However, it is stated explicitly in [2] that, during the authentication procedure, the authentication server will verify the correctness of the format of a provided identifier. Although, as in [3], the format of an identifier is not specified in [2], if the format is chosen with care (e.g. adhering to the rule suggested above) then this attack will not work.

- The use of a fingerprint measurement sample alone to generate the random number r appears to be an extremely dangerous procedure, especially if the fingerprint reader is not built into the card. We start by making this latter assumption.

  Suppose that a malicious interceptor E is able to observe two different (valid) authentication messages sent by the same card using the same value of r, but with different values of the time stamp T. The fact that the values of r are the same will be obvious to E since the values of C_1 will be the same. If, as we have assumed above, the fingerprint reader is not integral to the card, arranging for two identical fingerprint samples to be sent to a card should be relatively simple to achieve using a manipulated terminal.

Suppose these two messages are (i_A, C_1, C_2, T) and (i_A, C_1, C_2', T'). E now computes

$$\begin{aligned}
C_2.(C_2')^{-1} \bmod P &= (y_A \oplus h(p_A \oplus s_A))^r.(h(y_A \oplus T \oplus h(p_A \oplus s_A))). \\
&\quad (y_A \oplus h(p_A \oplus s_A))^{-r}. \\
&\quad (h(y_A \oplus T' \oplus h(p_A \oplus s_A)))^{-1} \bmod P \\
&= (h(y_A \oplus T \oplus h(p_A \oplus s_A))). \\
&\quad (h(y_A \oplus T' \oplus h(p_A \oplus s_A))) \bmod P.
\end{aligned}$$

Next suppose that E manages to steal the card used to generate these two messages, and also that E has the means to input to the card a correct fingerprint sample for user A. (Means by which this might be possible are discussed above — in particular, if the fingerprint reader is not integral to the card, then this would appear to be relatively simple to achieve).

E can now conduct an exhaustive search for p_A using the stolen card, without sending any messages to the authentication server. To check one guess for the password, p_A^* say, E simply uses the card to generate two authentication messages, both using the same value of r (readily achieved using an external fingerprint reader) and p_A^*, and where the timestamps used are the same as for the two previously intercepted messages. Suppose the resulting authentication messages are (i_A, C_1, D_2, T) and (i_A, C_1, D_2', T'). E now computes $D_2.(D_2')^{-1} \bmod P$ and compares it with the value $C_2.(C_2')^{-1} \bmod P$. If the guess for the password is correct then these values will be equal; if not then the values will almost certainly be different. This thus forms the basis for an exhaustive password search for a stolen card. If p_A is poorly chosen, as is usually the case for human-selected passwords, such an exhaustive search should be feasible, at least for some cards.

Of course, if the fingerprint reader is built into the card, then arranging for authentication messages to be generated using precisely the same value of r will be considerably more difficult. However, as discussed for example in [6, 7], with current card technology performing feature extraction from a fingerprint sample on a card is not really feasible; hence even if the fingerprint reader is integrated into the card, some of the processing associated with matching the sample to a template will need to be performed off the card. As a result, it may be possible for a malicious terminal to manipulate the value of r even if the fingerprint reader is built into the card. Finally we note that the authors of [3] do not appear to have done any experiments to justify their assertion that using a fingerprint sample to generate r is a technique that generates sufficient randomness to avoid the possibility of repeating values of r.

- The password change procedure is very hazardous, since the card does not have a means to check the validity of the current password. That is, the card password can be changed without entering the correct current password. This could give rise to problems as follows.
  - A user who follows the password change process and enters the incorrect value for the current password, e.g. in error, will render the card unusable.
  - If a malicious third party has temporary access to a card, and can somehow enter a correct biometric sample, then this party can change the password and thereby render the card unusable, even if this party does not know the correct password. A similar attack could be achieved by a malicious terminal, when the card is being used by the authorised user.

4 Concluding remarks

We have analysed a proposed remote user authentication scheme based on a smart card and biometrics. We have shown that the system fails to meet its objectives under reasonable assumptions about the threat model. In particular, if the biometric sensor (in this case a fingerprint reader) is not integrated into the card then serious attacks appear to be relatively simple to conduct.

Finally we note that the paper [3] describing the system analysed here does not contain a threat model. The absence of any such model may help to explain why the attacks described in this paper are possible, in that designing a system to be robust against feasible attacks does require a thorough understanding of what attacks may be possible.

References

[1] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, **IT-31**:469–472, 1985.

[2] J. K. Lee, S. R. Ryu, and K. Y. Yoo. Fingerprint-based remote user authentication scheme using smart cards. *Electronics Letters*, 38:554–555, 2002.

[3] C.-H. Lin and Y.-Y. Lai. A flexible biometrics remote user authentication scheme. *Computer Standards & Interfaces*, 27:19–23, 2004.

[4] T. Matsumoto. Gummy and conductive silicone rubber fingers. In Y. Zheng, editor, *Advances in Cryptology — ASIACRYPT 2002, 8th International Conference on the Theory and Application of Cryptology and Information Security, Queenstown, New Zealand, December 1-5, 2002, Proceedings*, number 2501 in Lecture Notes in Computer Science, pages 574–575. Springer-Verlag, Berlin, 2002.

[5] T. Matsumoto, H. Matsumoto, K. Yamada, and S. Hoshino. Impact of artificial "gummy fingers" on fingerprint systems. In R. L. van Renesse, editor, *Optical Security and Counterfeit Deterrence Techniques IV, Proceedings of SPIE Vol. 4677*, pages 275–289. SPIE: The International Society for Optical Engineering, 2002.

[6] L. Rila and C. J. Mitchell. Security analysis of smartcard to card reader communications for biometric cardholder authentication. In *Proceedings of CARDIS '02, 5th Smart Card Research and Advanced Application Conference, San Jose, California, November 2002*, pages 19–28. USENIX Association, Berkeley, CA, 2002.

[7] L. Rila and C. J. Mitchell. Security protocols for biometrics-based cardholder authentication in smartcards. In J. Zhou, M. Yung, and Y. Han, editors, *Applied Cryptography and Network Security - First International Conference, ACNS 2003 — Kunming, China, October 16-19 2003 — Proceedings*, volume 2846 of *Lecture Notes in Computer Science*, pages 254–264. Springer-Verlag, Berlin, 2003.