Cryptanalysis of a hybrid authentication protocol for large mobile networks

Qiang Tang *, Chris J. Mitchell

Information Security Group, Royal Holloway, University of London, Egham, Surrey TW20 0EX, UK

Received 31 August 2004; received in revised form 10 April 2005; accepted 10 April 2005
Available online 1 June 2005

Abstract

In this paper we analyse a hybrid authentication protocol due to Chien and Jan, designed for use in large mobile networks. The proposed protocol consists of two sub-protocols, namely the intra-domain authentication protocol and the inter-domain authentication protocol, which are used depending on whether the user and the request service are located in the same domain. We show that both sub-protocols suffer from a number of security vulnerabilities.

© 2005 Elsevier Inc. All rights reserved.

Keywords: Mobile security; Authentication; Public key cryptography; Key distribution

1. Introduction

With recent rapid development in computer network technologies, especially in mobile network technology, it has become easier and easier for people to access network services provided by a variety of service providers all over the world. Accordingly, a lot of research has been devoted to the authentication protocols which enable the users to be authenticated by the service providers before consuming the requested services—see for example Mitchell (2004). Among these existing authentication protocols, Kerberos, which was developed in the mid-80s as part of MIT’s Project Athena (Champine et al., 1990), is one of the most widely deployed protocols. Kerberos version 5 (Kohl and Neuman, 1993) is the current standard version. Although Kerberos is widely used, it is not only vulnerable to password guessing attacks but also very inefficient when inter-domain authentications are required. Many efforts have been devoted to improve the security, the scalability, and/or the efficiency of Kerberos, including Shieh et al. (1999), Kao and Chow (1995), Ganesan (1995), Fox and Gribble (1996), Sirbu and Chuang (1997), Samarakoon and Honary (1999), and Chien and Jan (2003).

In Chien and Jan (2003), Chien and Jan first demonstrate the security weaknesses in certain session key certificate based protocols (Kao and Chow, 1995; Shieh et al., 1999), and then propose a hybrid authentication protocol for large mobile networks based on public key cryptography, challenge-response and hash chaining. The proposed protocol consists of two sub-protocols, namely the intra-domain authentication protocol and the inter-domain authentication protocol, which are used depending on whether or not the user and the request service are located in the same domain. In the inter-domain authentication protocol, the user and the request service are located in the domain of the same KDC. In the inter-domain authentication protocol, it is assumed that each domain has a KDC and the KDC acts as the authority center for its domain. These different KDCs are organized as a DNS-based PKI tree hierarchy (Zao et al., 1999).

The authors Chien and Jan (2003) claim that their protocol simultaneously possesses several practical...
merits including good scalability, low communication and computational costs, and resistance to session key compromise attacks. However, we show that the proposed protocol suffers from a number of security problems.

The remainder of this paper is organised as follows. In Section 2, we review the proposed hybrid authentication protocol. In Section 3, we give our attacks on the proposed protocol. In Section 4, we describe the possible improvements and conclude the paper.

2. Review of the hybrid authentication protocol

The hybrid authentication protocol proposed in Chien and Jan (2003) provides both intra-domain and inter-domain authentication. The intra-domain authentication protocol is designed for an environment where all the users and servers are registered at one common key distribution centre, while the inter-domain protocol is for an environment with more than one key distribution centre. Both protocols are composed of two phases: initial authentication and subsequent authentication.

It is assumed that every principal, i.e. every user, server and KDC, possesses an asymmetric key pair which can be used for encrypting and decrypting data strings, and that every principal possesses a certificate for their public key signed by a generally trusted CA. Moreover, KDCs are assumed to possess personal information about each principal in their domain and be able to verify the certificate of each principal in their domain. To simplify matters we implicitly assume that the same key pair is used for both encryption and signature generation, although changing this assumption would be simple.

The following notation is used in the description of the hybrid authentication protocol.

- \( U, U_{ID} \): \( U \) is a user, and his identity is denoted by \( U_{ID} \).
- \( S, S_{ID} \): \( S \) is a server, and his identity is denoted by \( S_{ID} \).
- \((M)_K\): The result of symmetrically encrypting \( M \) using the secret key \( K \).
- \((M)_{Pub_X}\): The result of asymmetrically encrypting \( M \) using \( X \)'s public key \( Pub_X \).
- \( Cert_X \): The public key certificate of principal \( X \).

2.1. The intra-domain authentication protocol

We suppose that the Key Distribution Centre for the domain is KDC, and that \( S \) is a server registered with this KDC. If a user \( U \) wants to authenticate himself to \( S \), he initiates the following sub-protocols.

2.1.1. Initial authentication

1. \( U \rightarrow S: U_{ID}, \{N_U\}_{Pub_{KDC}}, Cert_{U_{ID}} \)

   \( U \) selects a random number \( N_U \) and encrypts it with the public key \( Pub_{KDC} \) of KDC. Then \( U \) sends his identity \( U_{ID} \), the encrypted nonce \( \{N_U\}_{Pub_{KDC}} \) and his public key certificate \( Cert_{U_{ID}} \) to \( S \).

2. \( S \rightarrow KDC: U_{ID}, \{N_U\}_{Pub_{KDC}}, Cert_{U_{ID}}, \{N_S\}_{Pub_{KDC}}, Cert_{S_{ID}} \)

   \( S \) selects a random number \( N_S \) and encrypts it with the public key \( Pub_{KDC} \) of KDC, then forwards the received data as well as his identity \( S_{ID} \), his public key certificate \( Cert_{S_{ID}} \) and his encrypted nonce \( \{N_S\}_{Pub_{KDC}} \) to KDC.

3. \( KDC \rightarrow S: \{U_{ID}, N_S, K, f^m(a), m, \{N_{U, S_{ID}}, a, f^m(a), m, Ticket_{U, S}\}_{Pub_{U}}\}_{Pub_{S}} \)

   KDC verifies the received certificates and, if the verification succeeds, decrypts \( \{N_S\}_{Pub_{KDC}} \) and \( \{N_U\}_{Pub_{KDC}} \). Then KDC chooses a random number \( a \) and a new master key \( K \) to be used by \( U \) and \( S \), and prepares a ticket \( Ticket_{U, S} = U_{ID}\|S_{ID}\|K\|VT\|Sig \) for this request, where VT is the validity period of this ticket, and \( Sig \) is KDC's signature on this ticket. Finally, KDC generates and sends the above message to \( S \), where \( f^m \) represents \( m \) iterations of hash-function \( f \), \( m \) is the maximum number of times that this ticket can be used.

4. \( S \rightarrow U: \{N_U, S_{ID}, a, f^m(a), m, Ticket_{U, S}\}_{Pub_{U}} \)

   \( S \) decrypts the message and checks the presence of the nonce \( N_S \) and \( U_{ID} \). If the check succeeds, he accepts this message and stores the values \( f^m(a) \) and \( m \) for later authentications and computes \( K_0 = f(K \oplus f^m(a)) \) as the first session key. \( S \) then discards the master key \( K \) and sends the above message to \( U \).

   \( U \) decrypts the received message and checks the nonce \( N_U \) and the ticket. If the check succeeds, he accepts this ticket and secretly stores \( a \) and \( K \). Then \( U \) computes and stores \( K_0 = f(K \oplus f^m(a)) \) and \( \{U_{ID}, Ticket_{U, S}\}_{Pub_{S}} \) for later authentications.

2.1.2. Subsequent authentication

In the \( i \)th subsequent authentication (\( 1 \leq i \leq m \)), \( U \) starts the following protocol.

\[ \text{Note: } S_{ID} \text{ is already contained in } Ticket_{U, S}, \text{ and thus two copies of } S_{ID} \text{ are present in } \{N_U, S_{ID}, a, f^m(a), m, Ticket_{U, S}\}_{Pub_{U}}. \text{ One copy of } S_{ID} \text{ can thus be deleted, in which case the encrypted message becomes } \{N_U, a, f^m(a), m, Ticket_{U, S}\}_{Pub_{U}}. \text{ If such a change is made, then the message in next step should be changed accordingly.} \]

\[ \text{This specification for message 4 differs slightly from the specification in Chien and Jan (2003), where it is stated that } S \text{ sends } \{N_U, S_{ID}, a, Ticket_{U, S}\}_{Pub_{U}}. \text{ This change has been made because the specification in Chien and Jan (2003) would appear to be an error, since } S \text{ cannot construct the message as specified in Chien and Jan (2003).} \]
1. $U \rightarrow S$: $\{U_{ID}, Ticket_{U,S}\}_{Pub_{S}}$, $(f^{m-1}(a))_{K_{i-1}}$
   $U$ sends the pre-computed data $\{U_{ID}, Ticket_{U,S}\}_{Pub_{S}}$ and $(f^{m-1}(a))_{K_{i-1}}$ to $S$.

2. $S \rightarrow U$: $(f^{m-1}(a))_{K_{i}}$
   $S$ decrypts $\{U_{ID}, Ticket_{U,S}\}_{Pub_{S}}$ to obtain the ticket $Ticket_{U,S}$. Using the information in the ticket, $S$ derives the master key $K$ and computes the current session key $K_{i-1} = f(K \oplus f^{m-1}(a))$, where $f^{m-1}(a)$ is the current stored hash value for $U$. He then uses this session key to decrypt the second part of the message, derives $f^{m-1}(a)$, and checks whether $f(f^{m-1}(a))$ equals the stored hash value $f^{m-1}(a)$. If the check succeeds, $S$ computes the new session key $K_{i} = f(K \oplus f^{m-1}(a))$ and sends $(f^{m-1}(a))_{K_{i}}$ to $U$. Finally, $S$ replaces the stored hash value with $f^{m-1}(a)$ and discards the ticket.

$U$ generates and uses the new session key $K_{i}$ to decrypt the received message and checks whether $f^{m-1}(a)$ is present. If so, he believes that $S$ has confirmed the new session key.

2.2. The inter-domain authentication protocol

Suppose a user $U_{X}$ wants to access the server $S_{Y}$, where $U_{X}$ is registered at KDC$_{X}$, $S_{Y}$ is registered at KDC$_{Y}$, and both KDC$_{X}$ and KDC$_{Y}$ are registered at KDC$_{0}$. $U_{X}$ initiates the following sub-protocols.

2.2.1. Initial authentication

1. $U_{X} \rightarrow S_{Y}$: $U_{X,ID}$, $\{N_{U_{X}}\}_{Pub_{KDC_{X}}}$, $Cert_{U_{X,ID}}$
   $U_{X}$ selects a random number $N_{U_{X}}$ and encrypts it with the public key $Pub_{KDC_{X}}$ of KDC$_{X}$. Then $U_{X}$ sends his identity $U_{X,ID}$, the encrypted nonce $\{N_{U_{X}}\}_{Pub_{KDC_{X}}}$ and his public key certificate $Cert_{U_{X,ID}}$ to $S_{Y}$.

2. $S_{Y} \rightarrow KDC_{Y}$: $U_{X,ID}$, $\{N_{U_{X}}\}_{Pub_{KDC_{X}}}$, $Cert_{U_{X,ID}}$, $S_{Y,ID}$, $\{N_{S_{Y}}\}_{Pub_{KDC_{Y}}}$, $Cert_{S_{Y,ID}}$
   $S_{Y}$ selects a random number $N_{S_{Y}}$ and encrypts it with the public key $Pub_{KDC_{Y}}$ of KDC$_{Y}$, then sends the received data as well as his identity $S_{Y,ID}$, his public key certificate $Cert_{S_{Y,ID}}$ and his encrypted nonce $\{N_{S_{Y}}\}_{Pub_{KDC_{Y}}}$ to KDC$_{Y}$.

3. KDC$_{Y}$ → KDC$_{0}$: $U_{X,ID}$, $\{N_{U_{X}}\}_{Pub_{KDC_{X}}}$, $Cert_{U_{X,ID}}$, $Cert_{KDC_{Y}}$, $\{N_{KDC_{Y}}\}_{Pub_{KDC_{0}}}$
   KDC$_{Y}$ decrypts $\{N_{S_{Y}}\}_{Pub_{KDC_{Y}}}$ and stores $N_{S_{Y}}$, selects a random number $N_{KDC_{Y}}$, and sends the above message to KDC$_{0}$.

4. KDC$_{0}$ → KDC$_{X}$: $U_{X,ID}$, $\{N_{U_{X}}\}_{Pub_{KDC_{X}}}$, $Cert_{U_{X,ID}}$, $S_{Y,ID}$, $\{N_{KDC_{Y}}\}_{Pub_{KDC_{X}}}$, $Cert_{KDC_{Y}}$
   KDC$_{0}$ decrypts $\{N_{KDC_{Y}}\}_{Pub_{KDC_{0}}}$, then generates and sends the above message to KDC$_{X}$, where KDC$_{Y,ID}$ denotes an identifier for KDC$_{Y}$.

5. KDC$_{X}$ → KDC$_{Y}$: $\{N_{KDC_{Y}}, S_{Y,ID}, U_{X,ID}, Cert_{U_{X,ID}}, info_{U_{X,ID}}, N_{U_{X}}\}_{Pub_{KDC_{Y}}}$
   KDC$_{Y}$ decrypts the received message and generates the personal information $info_{U_{X,ID}}$ regarding $U_{X}$, and sends the above message to KDC$_{Y}$. The personal information $info_{U_{X,ID}}$ consists of the validity period and privileges of $U_{X}$.

6. KDC$_{Y}$ → $S_{Y}$: $\{S_{Y}, U_{X,ID}, TID_{U,X}, K, f^{m}(a), m, \{N_{U_{X}}, S_{Y,ID}, Cert_{S_{Y,ID}}, TID_{U,X}, f^{m}(a), m, a, Ticket_{U_{X}}\}_{Pub_{S}}\}$
   KDC$_{Y}$ decrypts and checks $N_{KDC_{Y}}$ and $info_{U_{X,ID}}$. If the check succeeds, he assigns a temporary identity $TID_{U,X}$ for user $U_{X}$ and signs a ticket $Ticket_{U_{X}}$ for $U_{X}$, where the ticket has the same contents as in the intra-domain protocol except that $U_{ID}$ is replaced by $TID_{U,X}$ and $S_{ID}$ is replaced by $S_{Y,ID}$. Then KDC$_{Y}$ sends the above message to $S_{Y}$.

7. $S_{Y} \rightarrow U_{X}$: $\{N_{U_{X}}, S_{Y,ID}, Cert_{S_{Y,ID}}, TID_{U,X}, f^{m}(a), m, a, Ticket_{U_{X}}\}_{Pub_{S}}$
   $S_{Y}$ decrypts the received message, checks $N_{S_{Y}}$, computes $K_{0} = f(K \oplus f^{m}(a))$, and keeps $f(a)^{m}$ and $a$ for later authentication. Then $S_{Y}$ forwards the above message to $U_{X}$.

$U_{X}$ decrypts the received message and checks the nonce $N_{U_{X}}$ as well as the derived ticket. If the check succeeds, he accepts this ticket and stores $a$ and $K$. Then $U_{X}$ computes and stores $K_{0} = f(K \oplus f^{m}(a))$ and uses the ticket $Ticket_{U_{X}}$ for later authentications.

2.2.2. Subsequent authentication

The subsequent authentication procedure is identical to that in the intra-domain authentication protocol, except that $U_{ID}$ is replaced by $TID_{U,X}$.

3. Cryptanalysis results

We now show that the proposed scheme suffers from two serious security problems.

(1) The initial authentication part of the intra-domain authentication protocol has a major weakness. This allows a malicious but genuine user, $V$ say, who can interfere with messages sent and received by $S$, to impersonate another user, say $U$, to server $S$. The attack operates as follows.

(a) $V \rightarrow S$: $U_{ID}$, $\{N_{U}\}_{Pub_{KDC}}$, $Cert_{U_{ID}}$
   $V$ (pretending to be $U$) sends the first message of the initial authentication procedure to $S$.

---

3 This specification for message 6 differs slightly from the specification in Chien and Jan (2003), where it is stated that $S$ sends $\{N_{S_{Y}}, U_{X,ID}, TID_{U,X}, K, f^{m}(a), m, \{N_{U_{X}}, S_{Y,ID}, Cert_{S_{Y,ID}}, TID_{U,X}, a, Ticket_{U_{X}}\}_{Pub_{S}}\}$. This change has been made because the specification in Chien and Jan (2003) would appear to be an error, since otherwise $S$ cannot construct message 7 as specified in Chien and Jan (2003).
V first assembles the following dummy ticket, $Ticket_{U,S}$ as:

$$Ticket_{U,S} = U_ID||S_ID||K'||VT||Sig$$

where $Sig$ is a dummy signature (e.g. a random bit string of the right length). V then sends the first message of the subsequent authentication procedure (impersonating U) as:

(a) $V \rightarrow S$:  $\{U_ID, Ticket_{U,S}\}_Pub_V, (f^{m-1}(a'))_{K_{e-1}}$

Whether or not this is accepted by S as a valid message from U depends on how the message is processed by S. In the protocol description in Chien and Jan (2003) there is no mention of the checking of the signature $Sig$. If the description in Chien and Jan (2003) is followed, then this impersonation of U by V will be successful. However, checking of $Sig$ will reveal the fraud, and hence it is simple to repair this part of the protocol.

Finally note that a similar approach to that described above can be used by a malicious user $Y$ to learn the value of $N_U$ chosen by another user. It is not clear how this might be used to attack the protocols, but it does appear to be an undesirable feature (it also contradicts an assertion made in Section 4.1.1 of Chien and Jan (2003)).

(2) The initial authentication part of the inter-domain authentication protocol has a major weakness. This allows a malicious but genuine user, $V$, who can interfere with messages sent and received by other entities, to grant himself any privilege to access a server, regardless of whether or not $V$ should legitimately possess such a privilege. The attack operates as follows.

Suppose a user $V$ is registered at KDC$_X$ with identity $V_{ID}$, the $X$ is registered at KDC$_Y$, and that both KDC$_X$ and KDC$_Y$ are registered with KDC$_0$. Suppose further that $V$ is also registered at KDC$_0$ with identity $V_{ID}'$, and that server $S_0$ is registered with KDC$_0$. Note that we are assuming that KDC$_0$ is used both to certify lower level CAs (KDC$_X$ and KDC$_Y$), and to certify users and register servers—this is certainly not ruled out by Chien and Jan (2003).

To conduct the attack, $V$ first initiates the initial authentication of the inter-domain protocol with $S_Y$ as follows.

(a) $V \rightarrow S_Y$: $V_{ID}, \{N_Y\}_Pub_{KDC_Y}, Cert_{V_{ID}}$

$V$ sends the first message of the initial authentication procedure to $S_Y$.

(b) $S_Y \rightarrow KDC_Y$:  $V_{ID}, \{N_Y\}_Pub_{KDC_Y}, Cert_{V_{ID}}, \{N_{S_Y}\}_Pub_{KDC_Y}, Cert_{S_{ID}}$

$S_Y$ proceeds by sending the second message of the initial authentication procedure to KDC$_Y$.

(c) $KDC_Y \rightarrow KDC_X$:  $V_{ID}, \{N_Y\}_Pub_{KDC_X}, Cert_{V_{ID}}, \{N_{KDC_X}\}_Pub_{KDC_X}, Cert_{K_{ID}}, S_{ID}$

KDC$_Y$ proceeds by sending the third message of the initial authentication procedure to KDC$_0$. \[Q. Tang, C.J. Mitchell / The Journal of Systems and Software 79 (2006) 496–501\]

The above attack shows how it is possible to defeat the initial authentication procedure for the intra-domain protocol. We now show how, in certain circumstances, the above attack can be extended to the subsequent authentication procedure.
We suppose that this message is intercepted by V, and does not reach KDC0.

(d) V then starts an invocation of the intra-domain initial authentication procedure with server S0, using his second identity V' ID. Note that use of this procedure is appropriate since both V and S0 are registered with KDC0.

(i) \( V \rightarrow S0; \ V' \ ID; \ \{N_{KDCY} \}_{pbh_{KDC}}, \ Cert_{V'' ID} \)

Note that, rather than choosing a new random nonce \( N_V \) and encrypting it using the public key of KDC0, V copies the encrypted value of \( N_{KDCY} \) from the message KDC Y sent to KDC0 (in step c).

(ii) \( S0 \rightarrow KDC0; \ V' \ ID; \ \{N_{KDCY} \}_{pbh_{KDC}}, \ Cert_{V'' ID}, \ \{S00 \}_{pbh_{KDC}}, \ Cert_{S00}, \ S0 \)

S0 proceeds by sending the second message of the initial authentication procedure to KDC Y.

(iii) \( KDCY \rightarrow S0; \ \{V' \ ID, \ N_{KDCY}, K''m, f''(a''), m, \ \{N_{KDCY} \}_{S00}, \ N_{S00}, \ a'', f''(a''), m, Ticket_{V'' ID} \} \_{pbh} \)

KDC Y responds to S0 with the third message of the initial authentication procedure.

(iv) \( S0 \rightarrow V; \ \{N_{KDCY}, S00, a'', f''(a''), m, Ticket_{V'' ID} \} \_{pbhV} \)

S0 now sends the fourth message of the initial authentication procedure to V.

When V decrypts the received message, V gains a copy of \( N_{KDCY} \), which V should not know.

(e) \( V \rightarrow KDCY; \ \{N_{KDCY}, S'' ID, \ V' \ ID, \ Cert_{V'' ID}, \ Info_{V'' ID}, \ N_V \} \_{pbh_{KDCY}} \)

Using knowledge of \( N_{KDCY} \), V impersonates KDC Y to generate and send the message to KDC Y. It should be noted that V can set any valid time and privilege in \( Info_{V'' ID} \).

(f) \( KDCY \rightarrow S_Y; \ \{N_{S_Y}, S'' ID, TID_Y, K, f''(a''), m, \ \{N_{Y}, S'' ID, Cert_{S'' ID}, TID_Y, a, Ticket_{S'' ID} \} \_{pbh} \} \_{pbhV} \)

KDC Y decrypts and checks \( N_{KDCY} \) and \( info_{V'' ID} \). Since \( N_{KDCY} \) is correctly involved, the check will succeed. KDC Y assigns a temporary identity \( TID_Y \) and signs a ticket \( Ticket_{V'' S_Y} \) for V. Then KDC Y sends the above message to \( S_Y \).

(g) \( S_Y \rightarrow V; \ \{N_V, S'' ID, Cert_{S'' ID}, TID_Y, f''(a'')m, a, Ticket_{V'' S_Y} \} \_{pbhV} \)

\( S_Y \) decrypts the received message, checks \( N_{S_Y} \), computes \( K_Y = f(K \oplus f''(a'')) \), and keeps \( f(a)m \) and \( a \) for later authentication. Then \( S_Y \) forwards the above message to \( V \).

The above attack shows how it is possible to defeat the initial authentication procedure for the inter-domain protocol. Since all the authentication data is created correctly, even the signature in the ticket \( Ticket_{V'' S_Y} \) is also valid. So defeating the subsequent authentication in the inter-domain protocol is straightforward, and \( V \) is able to fraudulently obtain the service he wants.

4. Conclusions

In this paper we have analysed a hybrid authentication protocol designed for use in large mobile networks. We have shown that the proposed protocol suffers from a number of security problems.

Instead of time-stamps, the Chien–Jan protocol uses nonces to prevent replay attacks; however, this, combined with protocol design shortcomings, results in the security vulnerabilities in Section 3. To eliminate these vulnerabilities, we could require the KDCs to sign every message they send out. In addition, the server should validate the ticket \( Ticket_{U,S} \) the first time it receives it. These changes prevent the attacks identified in this paper; however, other attacks may still be possible. In general, it would be unwise to use this modified protocol, or any other protocol for that matter, without firm evidence of its robustness, e.g. as provided by a formal proof of security.

In the proposed protocol, public key cryptographic techniques are used for authentication, and the initial authentication phase needs to be re-executed when the hash chain is used up. For a mobile device with very limited resources, the associated computational requirements might be an unacceptably heavy burden. Improving the efficiency of the Chien–Jan protocol, whilst ensuring that it is secure, is a challenging task.

Acknowledgement

The authors would like to express deep appreciation to the reviewers for their valuable comments.

References


