Cryptanalysis of a hybrid authentication protocol for large mobile networks

Qiang Tang* and Chris J. Mitchell
Information Security Group
Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK

4th April 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.

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 [7]. Among these existing authentication protocols, Kerberos, which was developed in the mid-'80s as part of MIT's Project Athena [1], is one of

*Corresponding author. Email address: qiang.tang@rhul.ac.uk Telephone number: (+44) 1784414346 Fax number: (+44) 1784430766

the most widely deployed protocols. Kerberos version 5 [6] 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. [9], Kao and Chow [5], Ganesan [4], Fox and Gribble [3], Sirbu and Chuang [10], Samarakoon and Honary [8], and Chien and Jan [2].

In [2], Chien and Jan first demonstrate the security weaknesses in certain session key certificate based protocols [5, 9], 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 [11].

The authors [2] 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 [2] 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 pub-

lic 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}, $\{N_U\}_{Pub_{\mathrm{KDC}}}$, $Cert_{U_{\mathrm{ID}}}$

   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 $\{N_U\}_{Pub_{\mathrm{KDC}}}$ and his public key certificate $Cert_{U_{\mathrm{ID}}}$ to S.

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

   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 $Cert_{S_{\mathrm{ID}}}$ and his encrypted nonce $\{N_S\}_{Pub_{\mathrm{KDC}}}$ to KDC.

3. KDC $\rightarrow S$: $\{U_{\mathrm{ID}}, N_S, K, f^m(a), m, \{N_U, S_{\mathrm{ID}}, a, f^m(a), m, Ticket_{U,S}\}_{Pub_U}\}_{Pub_S}$ [1]

[1] Note that S_{ID} is already contained in $Ticket_{U,S}$, and thus two copies of S_{ID} are present

KDC verifies the received certificates and, if the verification succeeds, decrypts $\{N_S\}_{Pub_{\mathrm{KDC}}}$ and $\{N_U\}_{Pub_{\mathrm{KDC}}}$. Then KDC chooses a random number a and a new master key K to be used by U and S, and prepares a ticket $Ticket_{U,S} = U_{\mathrm{ID}}||S_{\mathrm{ID}}||K||VT||Sig$ 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. $S \rightarrow U$: $\{N_U, S_{\mathrm{ID}}, a, f^m(a), m, Ticket_{U,S}\}_{Pub_U}$ [2]

   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 $\{U_{\mathrm{ID}}, Ticket_{U,S}\}_{Pub_S}$ for later authentications.

2.1.2 Subsequent authentication

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

1. $U \rightarrow S$: $\{U_{\mathrm{ID}}, Ticket_{U,S}\}_{Pub_S}$, $(f^{m-i}(a))_{K_{i-1}}$

   U sends the pre-computed data $\{U_{\mathrm{ID}}, Ticket_{U,S}\}_{Pub_S}$ and $(f^{m-i}(a))_{K_{i-1}}$ to S.

2. $S \rightarrow U$: $(f^{m-i}(a))_{K_i}$

   S decrypts $\{U_{\mathrm{ID}}, Ticket_{U,S}\}_{Pub_S}$ 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 $K_{i-1} = f(K \oplus f^{m-i+1}(a))$, where $f^{m-i+1}(a)$ is the current stored hash value for U. He then uses this session key to decrypt the second part of the message, derives $f^{m-i}(a)$, and checks

in $\{N_U, S_{\mathrm{ID}}, a, f^m(a), m, Ticket_{U,S}\}_{Pub_U}$. One copy of S_{ID} can thus be deleted, in which case the encrypted message becomes $\{N_U, a, f^m(a), m, Ticket_{U,S}\}_{Pub_U}$. If such a change is made, then the message in next step should be changed accordingly.

[2] This specification for message 4 differs slightly from the specification in [2], where it is stated that S sends $\{N_U, S_{\mathrm{ID}}, a, Ticket_{U,S}\}_{Pub_U}$. This change has been made because the specification in [2] would appear to be an error, since S cannot construct the message as specified in [2].

whether $f(f^{m-i}(a))$ equals the stored hash value $f^{m-i+1}(a)$. If the check succeeds, S computes the new session key $K_i = f(K \oplus f^{m-i}(a))$ and sends $(f^{m-i}(a))_{K_i}$ to U. Finally, S replaces the stored hash value with $f^{m-i}(a)$ and discards the ticket.

U generates and uses the new session key K_i to decrypt the received message and checks whether $f^{m-i}(a)$ 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. $U_X \rightarrow S_Y$: $U_{X_{\mathrm{ID}}}$, $\{N_{U_X}\}_{Pub_{\mathrm{KDC}_X}}$, $Cert_{U_{X_{\mathrm{ID}}}}$
   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 $U_{X_{\mathrm{ID}}}$, the encrypted nonce $\{N_{U_X}\}_{Pub_{\mathrm{KDC}_X}}$ and his public key certificate $Cert_{U_{X_{\mathrm{ID}}}}$ to S_Y.
2. $S_Y \rightarrow \mathrm{KDC}_Y$: $U_{X_{\mathrm{ID}}}$, $\{N_{U_X}\}_{Pub_{\mathrm{KDC}_X}}$, $Cert_{U_{X_{\mathrm{ID}}}}$, $S_{Y_{\mathrm{ID}}}$, $\{N_{S_Y}\}_{Pub_{\mathrm{KDC}_Y}}$, $Cert_{S_{Y_{\mathrm{ID}}}}$
   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 $S_{Y_{\mathrm{ID}}}$, his public key certificate $Cert_{S_{Y_{\mathrm{ID}}}}$ and his encrypted nonce $\{N_{S_Y}\}_{Pub_{\mathrm{KDC}_Y}}$ to KDC_Y.
3. $\mathrm{KDC}_Y \rightarrow \mathrm{KDC}_0$: $U_{X_{\mathrm{ID}}}$, $\{N_{U_X}\}_{Pub_{\mathrm{KDC}_X}}$, $Cert_{U_{X_{\mathrm{ID}}}}$, $S_{Y_{\mathrm{ID}}}$, $Cert_{\mathrm{KDC}_Y}$, $\{N_{\mathrm{KDC}_Y}\}_{Pub_{\mathrm{KDC}_0}}$
   KDC_Y decrypts $\{N_{S_Y}\}_{Pub_{\mathrm{KDC}_Y}}$ and stores N_{S_Y}, selects a random number N_{KDC_Y}, and sends the above message to KDC_0.
4. $\mathrm{KDC}_0 \rightarrow \mathrm{KDC}_X$: $\{U_{X_{\mathrm{ID}}}, \{N_{U_X}\}_{Pub_{\mathrm{KDC}_X}}, Cert_{U_{X_{\mathrm{ID}}}}, S_{Y_{\mathrm{ID}}}, \mathrm{KDC}_{Y_{\mathrm{ID}}}, N_{\mathrm{KDC}_Y}, Cert_{\mathrm{KDC}_Y}\}_{Pub_{\mathrm{KDC}_X}}$
   KDC_0 decrypts $\{N_{\mathrm{KDC}_Y}\}_{Pub_{\mathrm{KDC}_0}}$, then generates and sends the above message to KDC_X, where $\mathrm{KDC}_{Y_{\mathrm{ID}}}$ denotes an identifier for KDC_Y.
5. $\mathrm{KDC}_X \rightarrow \mathrm{KDC}_Y$: $\{N_{\mathrm{KDC}_Y}, S_{Y_{\mathrm{ID}}}, U_{X_{\mathrm{ID}}}, Cert_{U_{X_{\mathrm{ID}}}}, info_{U_{X_{\mathrm{ID}}}}, N_{U_X}\}_{Pub_{\mathrm{KDC}_Y}}$
   KDC_X decrypts the received message and $\{N_{U_X}\}_{Pub_{\mathrm{KDC}_X}}$, generates the personal information $info_{U_{X_{\mathrm{ID}}}}$ regarding U_X, and sends the above message to KDC_Y. The personal information $info_{U_{X_{\mathrm{ID}}}}$ consists of the validity period and privileges of U_X.

6. $\text{KDC}_Y \rightarrow S_Y$: $\{N_{S_Y}, U_{X_{\text{ID}}}, TID_{U_X}, K, f^m(a), m, \{N_{U_X}, S_{Y_{\text{ID}}}, Cert_{S_{Y_{\text{ID}}}}, TID_{U_X}, f^m(a), m, a, Ticket_{U_X,S_Y}\}_{Pub_{U_X}}\}_{Pub_{S_Y}}$ [3]
   KDC_Y decrypts and checks N_{KDC_Y} and $info_{U_{X_{\text{ID}}}}$. If the check succeeds, he assigns a temporary identity TID_{U_X} for user U_X and signs a ticket $Ticket_{U_X,S_Y}$ 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 $S_{Y_{\text{ID}}}$. Then KDC_Y sends the above message to S_Y.

7. $S_Y \rightarrow U_X$: $\{N_{U_X}, S_{Y_{\text{ID}}}, Cert_{S_{Y_{\text{ID}}}}, TID_{U_X}, f^m(a), m, a, Ticket_{U_X,S_Y}\}_{Pub_{U_X}}$
   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 secretly stores a and K. Then U_X computes and stores $K_0 = f(K \oplus f^m(a))$ and $\{TID_{U_X}, Ticket_{U_X,S_Y}\}_{Pub_{S_Y}}$ 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}, $\{N_U\}_{Pub_{\text{KDC}}}$, $Cert_{U_{\text{ID}}}$
   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 [2], where it is stated that S sends $\{N_{S_Y}, U_{X_{\text{ID}}}, TID_{U_X}, K, f^m(a), m, \{N_{U_X}, S_{Y_{\text{ID}}}, Cert_{S_{Y_{\text{ID}}}}, TID_{U_X}, a, Ticket_{U_X,S_Y}\}_{Pub_{U_X}}\}_{Pub_{S_Y}}$. This change has been made because the specification in [2] would appear to be an error, since otherwise S cannot construct message 7 as specified in [2].

(b) $S \rightarrow$ KDC: U_{ID}, $\{N_U\}_{Pub_{\mathrm{KDC}}}$, $Cert_{U_{\mathrm{ID}}}$, S_{ID}, $\{N_S\}_{Pub_{\mathrm{KDC}}}$, $Cert_{S_{\mathrm{ID}}}$
S proceeds by sending the second message of the initial authentication procedure to KDC. We suppose that this message is intercepted by V, and does not reach KDC.

(c) V, now acting on his/her own behalf, starts a second invocation of the initial authentication procedure.

i. $V \rightarrow S$: V_{ID}, $\{N_S\}_{Pub_{\mathrm{KDC}}}$, $Cert_{V_{\mathrm{ID}}}$
Note that, rather than choosing a new random nonce N_V and encrypting it using the public key of KDC, V copies the encrypted value of N_S from the message S sent to KDC (in step b).

ii. $S \rightarrow$ KDC: V_{ID}, $\{N_S\}_{Pub_{\mathrm{KDC}}}$, $Cert_{V_{\mathrm{ID}}}$, S_{ID}, $\{N'_S\}_{Pub_{\mathrm{KDC}}}$, $Cert_{S_{\mathrm{ID}}}$
S proceeds by sending the second message of the initial authentication procedure to KDC.

iii. KDC $\rightarrow S$: $\{V_{\mathrm{ID}}, N'_S, K, f^m(a), m, \{N_S, S_{\mathrm{ID}}, a, f^m(a), m, Ticket_{V,S}\}_{Pub_V}\}_{Pub_S}$
KDC responds to S with the third message of the initial authentication procedure.

iv. $S \rightarrow V$: $\{N_S, S_{\mathrm{ID}}, a, f^m(a), m, Ticket_{V,S}\}_{Pub_V}$
S now sends the fourth message of the initial authentication procedure to V.

When V decrypts the received message, V has a copy of N_S, which V should not know. V can further recover a (and also K from $Ticket_{V,S}$). V can use this information to fabricate the third message of the first invocation of the initial authentication procedure (to make it look as if it comes from KDC), as follows. V generates K' and a', computes $f^m(a')$, and puts $Ticket_{U,S} = U_{\mathrm{ID}}||S_{\mathrm{ID}}||K'||VT||Sig$, where Sig is a random bit string of the right length. Then V impersonates KDC to send the following message to S. Observe that S has no way of knowing that the encrypted string within the message is encrypted under Pub_V rather than Pub_U.

$V \rightarrow S$: $\{U_{\mathrm{ID}}, N_S, K', f^m(a'), m, \{N_S, S_{\mathrm{ID}}, a', f^m(a'), m, Ticket_{U,S}\}_{Pub_V}\}_{Pub_S}$

(d) $S \rightarrow V$: $\{N_U, S_{\mathrm{ID}}, a', f^m(a'), m, Ticket_{U,S}\}_{Pub_V}$
When S decrypts the received message, the value of N_S will be correctly included, as is U_{ID}, at which point S will falsely believe that the first message (in step a) came from U. S will now send the final message of the initial authentication procedure to U, which we suppose that V suppresses.

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.

V first assembles the following *dummy* ticket, $Ticket_{U,S}$ as: $Ticket_{U,S} = U_{\mathrm{ID}}||S_{\mathrm{ID}}||K'||VT||Sig$ 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) $V \rightarrow S$: $\{U_{\mathrm{ID}}, Ticket_{U,S}\}_{Pub_S}$, $(f^{m-i}(a'))_{K_{i-1}}$

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 [2] there is no mention of the checking of the signature Sig. If the description in [2] 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 V 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 [2]).

2. The initial authentication part of the inter-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 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 S_Y 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 [2].

   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}, $\{N_V\}_{Pub_{\mathrm{KDC}_X}}$, $Cert_{V_{\mathrm{ID}}}$
   V sends the first message of the initial authentication procedure to S_Y.

(b) $S_Y \rightarrow \mathrm{KDC}_Y$: V_{ID}, $\{N_V\}_{Pub_{\mathrm{KDC}_X}}$, $Cert_{V_{\mathrm{ID}}}$, $S_{Y_{\mathrm{ID}}}$, $\{N_{S_Y}\}_{Pub_{\mathrm{KDC}_Y}}$, $Cert_{S_{Y_{\mathrm{ID}}}}$
S_Y proceeds by sending the second message of the initial authentication procedure to KDC_Y.

(c) $\mathrm{KDC}_Y \rightarrow \mathrm{KDC}_0$: $V_{\mathrm{ID}}, \{N_V\}_{Pub_{\mathrm{KDC}_X}}$, $Cert_{V_{\mathrm{ID}}}$, $S_{Y_{\mathrm{ID}}}$, $Cert_{\mathrm{KDC}_Y}$, $\{N_{\mathrm{KDC}_Y}\}_{Pub_{\mathrm{KDC}_0}}$
KDC_Y proceeds by sending the third message of the initial authentication procedure to KDC_0. We suppose that this message is intercepted by V, and does not reach KDC_0.

(d) V then starts an invocation of the **intra-domain** initial authentication procedure with server S_0, using his second identity V^*_{ID}. Note that use of this procedure is appropriate since both V and S_0 are registered with KDC_0.

i. $V \rightarrow S_0$: V^*_{ID}, $\{N_{\mathrm{KDC}_Y}\}_{Pub_{\mathrm{KDC}_0}}$, $Cert_{V^*_{\mathrm{ID}}}$
Note that, rather than choosing a new random nonce N_V and encrypting it using the public key of KDC_0, V copies the encrypted value of N_{KDC_Y} from the message KDC_Y sent to KDC_0 (in step c).

ii. $S_0 \rightarrow \mathrm{KDC}_0$: V^*_{ID}, $\{N_{\mathrm{KDC}_Y}\}_{Pub_{\mathrm{KDC}_0}}$, $Cert_{V^*_{\mathrm{ID}}}$, $S_{0_{\mathrm{ID}}}$, $\{N_{S_0}\}_{Pub_{\mathrm{KDC}_0}}$, $Cert_{S_{0_{\mathrm{ID}}}}$
S_0 proceeds by sending the second message of the initial authentication procedure to KDC_0.

iii. $\mathrm{KDC}_0 \rightarrow S_0$: $\{V^*_{\mathrm{ID}}, N_{S_0}, K'', f^m(a''), m, \{N_{\mathrm{KDC}_Y}, S_{0_{\mathrm{ID}}}, a'', f^m(a''), m, Ticket_{V,S}\}_{Pub_V}\}_{Pub_{S_0}}$
KDC_0 responds to S_0 with the third message of the initial authentication procedure.

iv. $S_0 \rightarrow V$: $\{N_{\mathrm{KDC}_Y}, S_{0_{\mathrm{ID}}}, a'', f^m(a''), m, Ticket_{V,S}\}_{Pub_V}$
S_0 now sends the fourth message of the initial authentication procedure to V.

When V decrypts the received message, V gains a copy of N_{KDC_Y}, which V should not know.

(e) $V \rightarrow \mathrm{KDC}_Y$: $\{N_{\mathrm{KDC}_Y}, S_{Y_{\mathrm{ID}}}, V_{\mathrm{ID}}, Cert_{V_{\mathrm{ID}}}, Info_{V_{\mathrm{ID}}}, N_V\}_{Pub_{\mathrm{KDC}_Y}}$
Using knowledge of N_{KDC_Y}, V impersonates KDC_X to generate and send the message to KDC_Y. It should be noted that V can set any valid time and privilege in $Info_{V_{\mathrm{ID}}}$.

(f) $\mathrm{KDC}_Y \rightarrow S_Y$: $\{N_{S_Y}, V_{\mathrm{ID}}, TID_V, K, f^m(a), m, \{N_V, S_{Y_{\mathrm{ID}}}, Cert_{S_{Y_{\mathrm{ID}}}}, TID_V, a, Ticket_{V,S_Y}\}_{Pub_V}\}_{Pub_{S_Y}}$
KDC_Y decrypts and checks N_{KDC_Y} and $info_{V_{\mathrm{ID}}}$. Since N_{KDC_Y} is correctly involved, the check will succeed. KDC_Y assigns a temporary identity TID_V and signs a ticket $Ticket_{V,S_Y}$ for V. Then KDC_Y sends the above message to S_Y.

(g) $S_Y \rightarrow V$: $\{N_V, S_{Y_{\mathrm{ID}}}, Cert_{S_{Y_{\mathrm{ID}}}}, TID_V, f^m(a), m, a, Ticket_{V,S_Y}\}_{Pub_V}$
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 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.

5 Acknowledgements

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

References

[1] G. A. Champine, D. E. Geer. Jr., and W. N. Ruh. Project Athena as a Distributed Computer System. *Computer*, 23(9):40–51, 1990.

[2] H. Y. Chien and J. K. Jan. A hybrid authentication protocol for large mobile network. *Journal of Systems and Software*, 67:123–137, 2003.

[3] A. Fox and S. D. Gribble. Security on the move: indirect authentication using Kerberos. In *Proceedings of the Second Annual International Conference on Mobile Computing and Networking*, pages 155–164. ACM Press, 1996.

[4] R. Ganesan. Yaksha: augmenting kerberos with public key cryptography. In *SNDSS '95: Proceedings of the 1995 Symposium on Network and Distributed System Security*, pages 132–143. IEEE Computer Society, 1995.

[5] I.-L. Kao and R. Chow. An efficient and secure authentication protocol using uncertified keys. *ACM Operating Systems Review*, 29(3):14–21, 1995.

[6] J. Kohl and C. Neuman. The Kerberos network authentication service (v5). Internet Request for Comments 1510, 1993.

[7] C. J. Mitchell, editor. *Security for Mobility*. IEE Press, London, 2004.

[8] M. I. Samarakoon and B. Honary. Novel authentication and key agreement protocol for low processing power and systems resource requirements in portable communications systems. In *IEE Colloquium on Novel DSP Algorithms and Architectures for Radio Systems*, pages 9/1–9/5, 1999.

[9] S. P. Shieh, F. S. Ho, and Y. L. Huang. An efficient authentication protocol for mobile networks. *Journal of Information Science and Engineering*, 15(4):505–520, 1999.

[10] M. A. Sirbu and J. C. I. Chuang. Distributed authentication in Kerberos using public key cryptography. In *Proceedings of the 1997 Symposium on Network and Distributed System Security*, pages 134–141. IEEE Computer Society, 1997.

[11] J. Zao, S. Kent, J. Gahm, G. Troxel, M. Condell, P. Helinek, N. Yuan, and I. Castineyra. A public-key based secure mobile IP. *Wireless Networks*, 5(5):373–390, 1999.