Revision and Enhancement of Two Three Party Key Agreement Protocols Vulnerable to KCI Attacks

Maurizio Adriano Strangio

Department of Mathematics, University of Rome "Roma Tre", Rome, Italy

*Corresponding Author: strangio@mat.uniroma3.it

Copyright ©2014 Horizon Research Publishing All rights reserved.

Abstract In two recent papers, Zuowen Tan (Security and Communication Networks) and Chih-ho Chou *et al.* (Computers and Electronics) published three party key agreement protocols especially convenient for protecting communications in mobile-centric environments such as e-payments, vehicular mobile networks (VMN), RFID applications, etc.

For his protocol, Tan provides a formal security proof developed in a model of distributed computing based on the seminal work of Bellare and Rogaway.

In this paper, we show that both protocols are vulnerable to KCI attacks. We suggest modifications to both protocols that fix the vulnerability at the expense of a small decrease in their computational efficiency.

Keywords Three Party Key Agreement, Key-compromise Impersonation, Mobile Network Communications

1 Introduction

In two recent papers, Tan [2] and Chou *et al.* [5] present three party key agreement protocols especially convenient for protecting communications in mobile-centric environments such as e-payments, vehicular mobile networks (VMN), RFID applications, etc.

For his protocol, Tan provides a formal security proof in a model of distributed computing based on the work of Bellare and Rogaway [6, 20] and Abdalla *et al.* [1].

Unfortunately, both protocols are vulnerable to a particular man-in-the-middle attack known as Key Compromise Impersonation (KCI) [12, 15]. In such attacks, an adversary that has obtained the private key of party A can impersonate a legitimate peer B ; if the attack is successful A will share a session key with the adversary (instead of B). This is a subtle attack that can have drastic consequences since the adversary may obtain personal information from A such as secret keying

material (e.g. passwords) or private data (e.g. credit card numbers). With three party protocols the adversary may impersonate a peer (A or B) or the trusted third party (S).

In this paper, we describe successful KCI attacks against the aforementioned protocols and also suggest modifications to fix the vulnerabilities at the expense of a small decrease in their computational efficiency.

The rest of the paper is organized as follows. We describe KCI attacks against the protocols of TAN and CHOU *et al.* respectively in Sections 2 and 3. In Section 4, we suggest modifications to the above protocols in order to prevent those attacks. Finally, Section 5 contains our concluding remarks.

2 A KCI attack against Tan's protocol

In this section we review Tan's three-party key agreement protocol and describe how a successful KCI attack can be conducted by an adversary that has compromised the private keying material of an honest party.

2.1 Review of the protocol specification

The protocol consists of an initialization phase (wherein A and B register with the trusted server S and obtain an $d_A = H(ID_A || x)$ and $d_B = H(ID_B || x)$ respectively, where $x \in F_q$ is the master key held by S) and the subsequent authenticated key agreement phase (Figure 1).

System parameters are defined by the following tuple:

$$\Phi_{TAN} = (p, n, E, G, P, E_k(\cdot), D_k(\cdot), h(\cdot))$$

were

- (i) p, n are prime numbers;
- (ii) $E(\mathbb{F}_p)$ is an elliptic curve defined by the equation $y^2 = x^3 + ax + b$ over the finite field $\mathbb{F}_p$, where $4a^3 + 27b^2 \neq 0 \pmod{p}$;

- (iii) master key $x \in_R \mathbb{Z}_n^*$, the later is the set of residues modulo n ;
- (iv) group G and point P with order n over $E(\mathbb{F}_p)$;
- (v) symmetric encryption/decryption algorithms $E_k(\cdot)$, $D_k(\cdot)$;
- (vi) hash function $h(\cdot)$.

$A[d_A], B[d_B], S[d_A, d_B, x]$	
$A \rightarrow B:$	$\{ID_A, \text{request}\}$
$A:$	$a \xleftarrow{R} \mathbb{Z}_n^*$ $R_1 \leftarrow aP, e_1 \leftarrow h(R_1 \ ID_A \ ID_B)$ $e_2 \leftarrow E_{d_A}(R_1 \ ID_A \ ID_B \ e_1)$
$A \rightarrow S:$	$\{ID_A, ID_B, e_2\}$
$B:$	$b \xleftarrow{R} \mathbb{Z}_n^*$ $R_2 \leftarrow bP, e_3 \leftarrow h(R_2 \ ID_B \ ID_A)$ $e_4 \leftarrow E_{d_B}(R_2 \ ID_B \ ID_A \ e_3)$
$B \rightarrow S:$	$\{ID_B, ID_A, e_4\}$
$S:$	$d_A \leftarrow h(ID_A \ x), d_B \leftarrow h(ID_B \ x)$ $R'_1 \ ID'_A \ ID'_B \ e'_1 = D_{d_A}(e_2)$ $R'_2 \ ID'_B \ ID'_A \ e'_3 = D_{d_B}(e_4)$ if $e'_1 \neq h(R'_1 \ ID_A \ ID_B)$ abort if $e'_3 \neq h(R'_2 \ ID_B \ ID_A)$ abort $e \leftarrow h(R'_1 \ R'_2 \ ID_A \ ID_B)$ $Q_1 \leftarrow E_{d_A}(R'_1 \ R'_2 \ e)$ $Q_2 \leftarrow E_{d_B}(R'_2 \ R'_1 \ e)$
$S \rightarrow A:$	$\{Q_1\}$
$S \rightarrow B:$	$\{Q_2\}$
$A:$	$R''_1 \ R''_2 \ e'' = D_{d_A}(Q_1)$ if $R_1 \neq R'_1$ abort if $e'' \neq R''_1 \ R''_2 \ ID_A \ ID_B$ abort $sk \leftarrow h(aR''_2 \ ID_A \ ID_B)$
$B:$	$R''_2 \ R''_1 \ e'' = D_{d_B}(Q_2)$ if $R_2 \neq R'_2$ abort if $e'' \neq R''_1 \ R''_2 \ ID_A \ ID_B$ abort $sk \leftarrow h(bR''_1 \ ID_A \ ID_B)$

Figure 1. TAN's Protocol

2.2 Description of the KCI attack scenario

Below we provide a detailed description of a successful KCI attack against Tan's protocol:

- Adversary $\mathcal{A}$ obtains A 's private key $d_A = h(ID_A \| x)$;
- A generates a random nonce $a \in \mathbb{Z}_n^*$, computes $R_1 = aP$, $e_1 = h(R_1 \| ID_A \| ID_B)$, $e_2 = E_{d_A}(R_1 \| ID_A \| ID_B \| e_1)$ and sends $\{ID_A, \text{request}\}$ and $\{ID_A, ID_B, e_2\}$ to B and S respectively to initiate a protocol instance with B ;
- $\mathcal{A}$ intercepts the messages $\{ID_A, \text{request}\}$ and $\{ID_A, ID_B, e_2\}$, decrypts the ciphertext $D_{d_A}(e_2) = R'_1 \| ID'_A \| ID'_B \| e'_1$ chooses a random nonce $c \in \mathbb{Z}_n^*$, computes $R_2 = cP$, $e = h(R'_1 \| R_2 \| ID_A \| ID_B)$, $Q_1 = E_{d_A}(R'_1 \| R_2 \| e)$, and sends Q_1 to A ;

- on receipt of Q_1 , A computes $D_{d_A}(Q_1) = R''_1 \| R''_2 \| e''$. If the equations $R_1 = R''_1$ and $e'' = R'_1 \| R''_2 \| ID_A \| ID_B$ are verified (indeed they are since the transcripts are indistinguishable from those exchanged by honest parties) A terminates with the session key $sk_A = h(aR''_2 \| ID_A \| ID_B)$;
- $\mathcal{A}$ computes $sk' = h(cR'_1 \| ID_A \| ID_B)$ and will be able to establish a communication session with A since $sk' = sk_A$.

3 A KCI attack against Chou *et al.*'s protocol

In this section we review Chou *et al.*'s three-party key agreement protocol and describe how a successful KCI attack can be conducted by an adversary.

3.1 Review of the protocol specification

This protocol also comprises an initialization phase (wherein users A and B register with the trusted server S and obtain $Y_A = sk_A PK_S$ and $Y_B = sk_B PK_S$) and an authenticated key agreement phase (Figure 2).

System parameters are defined by the following tuple:

$$\Phi_{CHOU} = (p, n, E, G, P, E_k(\cdot), D_k(\cdot), h(\cdot))$$

where each parameter is defined as in Section 2.

$A[sk_A, PK_A], B[sk_B, PK_B], S[sk_S, PK_S]$	
$A:$	$r_A \xleftarrow{R} \mathbb{Z}_p^*$ $R_A \leftarrow r_A PK_A + Y_A$ $K_A \leftarrow r_A Y_A = (K_{A.x}, K_{A.y})$ $C_A \leftarrow E_{K_{A.x}}(R_A, ID_A, ID_B, T_A)$
$A \rightarrow B:$	$\{ID_A, \text{request}\}$
$A \rightarrow S:$	$\{ID_A, R_A, C_A, T_A\}$
$B:$	$r_B \xleftarrow{R} \mathbb{Z}_p^*$ $R_B \leftarrow r_B PK_B + Y_B$ $K_B \leftarrow r_B Y_B = (K_{B.x}, K_{B.y})$ $C_B \leftarrow E_{K_{B.x}}(R_B, ID_B, ID_A, T_B)$
$B \rightarrow A:$	$\{ID_B, \text{response}\}$
$B \rightarrow S:$	$\{ID_B, R_B, C_B, T_B\}$
$S:$	$K_A \leftarrow sk_S(R_A - sk_S PK_A) = (K_{A.x}, K_{A.y})$ $K_B \leftarrow sk_S(R_B - sk_S PK_B) = (K_{B.x}, K_{B.y})$ $R_A, ID_A, ID_B, T_A = D_{K_{A.x}}(C_A)$ $R_B, ID_B, ID_A, T_B = D_{K_{B.x}}(C_B)$ check T_A, T_B, R_A, R_B $C_{SA} \leftarrow E_{K_{A.x}}(R_A, K_B, ID_A, ID_S, T_S)$ $C_{SB} \leftarrow E_{K_{B.x}}(R_B, K_A, ID_B, ID_S, T_S)$
$S \rightarrow A:$	$\{C_{SA}, T_S\}$
$S \rightarrow B:$	$\{C_{SB}, T_S\}$
$A:$	$R_A, K_B, ID_A, ID_S, T_S = D_{K_{A.x}}(C_{SA})$ check R_A, T_S $sk \leftarrow r_A sk_A K_B$
$B:$	$R_B, K_A, ID_B, ID_S, T_S = D_{K_{B.x}}(C_{SB})$ check R_B, T_S $sk \leftarrow r_B sk_B K_A$

Figure 2. Chou *et al.*'s Protocol

3.2 Description of the KCI attack scenario

Below we provide a detailed description of a successful KCI attack against Chou's protocol:

1. Adversary $\mathcal{A}$ obtains A 's private key sk_A ;
2. A executes all operations according to the protocol specification and sends $\{ID_A, \text{request}\}$ and $\{ID_A, R_A, C_A, T_A\}$ to B and S respectively to initiate a protocol instance with B ;
3. $\mathcal{A}$ intercepts the message $\{ID_A, R_A, C_A, T_A\}$, chooses a random nonce $r_E \in \mathbb{Z}_p^*$ and sends $C_{SA} = E_{K_{A,x}}(R_A, K_B, ID_B, ID_S, T_S)$ where $K_B = r_E P$ to A ;
4. on receipt of C_{SA} , A follows the protocol specification (R_A, T_S will pass the verification step) and terminates with the session key $sk = r_A sk_A K_B = r_A sk_A r_E P = r_E r_A P K_A = r_E (R_A - Y_A)$;
5. $\mathcal{A}$ can establish a communication session with A by computing $sk = r_E (R_A - Y_A)$ where $Y_A = sk_A P K_S$ and thus is easily computed by the adversary.

4 Revisiting the protocols to eliminate the vulnerability to KCI attacks

In this section we illustrate modifications to the protocols of Tan and Chou *et al.* that do not allow a malicious party to perform successful KCI attacks.

4.1 A KCI-resilient version of Tan's protocol

It is interesting to notice that Tan's protocol cannot withstand KCI attacks despite the fact that a formal security proof was provided by the author (see Theorems 1, 2 in [2]). The arguments used by the author to support the proof of KCI-resilience (Theorem 2) require that the adversary must be able to forge the transcript e_d ; however, this assumption misses the point since the adversary does not need to faithfully reproduce the protocol actions of B but must simply generate message transcripts (in this particular case, a Diffie-Hellman ephemeral key $R_2 = cP$) that are *indistinguishable* (for A) from the real ones. In general, for the sake of protocol security analysis one assumes that a man-in-the-middle attacker has total control over the network (i.e. she can insert, delete, modify messages flowing across the network) and is allowed to achieve her goals by arbitrarily diverging from the protocol specification.

To prevent the attack Tan's protocol can be modified as follows: the trusted third party S shall generate keys for each peer $D_A = d_A P, D_B = d_B P$ (eventually during the initialization stage) and compute $e = h(R'_1 \| R'_2 \| ID_A \| ID_B \| D_A \| D_B)$, $Q_1 = E_{d_A}(R'_1 \| R'_2 \| e)$ and $Q_2 = E_{d_B}(R'_2 \| R'_1 \| e)$; finally A shall compute its session key as $sk = h(aD_B \| d_A R'_2 \| ID_A \| ID_B)$ (similarly for B).

With the preceding modifications, Tan's key exchange can now withstand KCI attacks. Indeed, the session key is computed using the method introduced in the MTI/A0 [9] protocol, which is immune from KCI attacks; the adversary must now obtain both the private key d_A and the random nonce a to compute the session key of A . A significant difference with the MTI protocol is due to the fact that the keys D_A, D_B do not necessarily have to be pre-distributed to A, B respectively and therefore do not require public key certificates.

The revised protocol enjoys KCI attack resilience at the expense of an increased computational workload with respect to the original version. In particular, the trusted first party S must now compute two additional scalar multiplications to generate the keys D_A, D_B . However, for reasons of efficiency the computation can be performed prior to online executions of the protocol. Each peer A, B is required to compute an additional scalar multiplication (aD_B, bD_A) at runtime to compute the session key.

4.2 A KCI-resilient version of Chou *et al.*'s protocol

It is worth mentioning that the protocol of Chou *et al.* is actually a revised version of a flawed protocol previously proposed by Zuowen Tan [3] which was yet another version of the protocol originally published by Yang and Chang [4] (the later also contained a vulnerability to impersonation attacks).

Chou *et al.*'s protocol can withstand KCI attacks with the following modifications. The trusted third party computes $C_{SA} = E_{K_{A,x}}(R_A, K_B, Y_B, ID_A, ID_S, T_S)$, $C_{SB} = E_{K_{B,x}}(R_B, K_A, Y_A, ID_B, ID_S, T_S)$ and sends these transcripts to A, B respectively; with respect to the original specification the terms Y_B, Y_A are included in C_{SA}, C_{SB} respectively. On receipt of C_{SA} , A computes the session key $sk = h(r_A Y_B \| sk_A K_B)$ (similarly for B) where h is an appropriate hash function.

The revised protocol (similarly to Tan's protocol) enjoys KCI attack resilience at the expense of a slightly increased computational workload with respect to the original version. In this case, each peer A, B needs an additional scalar multiplication ($r_A Y_B, r_B Y_A, ID_A, ID_B$) at runtime to compute the session key.

5 Concluding remarks

We have shown that the three party key agreement protocols recently published in the literature by Tan [2] and Chou *et al.* [5] are not resilient to KCI attacks although their authors claim the contrary.

Designing secure key agreement protocols is far from being a simple task, such protocols involve so many details and complicated interactions between different cryptographic primitives that it is nearly impossible to establish beyond doubt that they are infallible. Indeed, in the history of this subject there is an abundance of protocols that were more or less trivially broken regardless of whether formal or heuristic arguments were provided to support security claims.

One of the main problems with formal security proofs

is that few people are capable of verifying them because the maths can be quite sophisticated and may be therefore difficult to understand for the average practitioner. In the last two decades many formal security models have been proposed (which often differ by a few details) but it still is not quite clear how to "transfer" a security property proved in one model to a different model and to the real world (Choo *et al.* [14] have examined this issue). Furthermore, provable security of cryptographic primitives is largely based on the use of computational assumptions, these have proliferated in the last decade [16] making it difficult to differentiate between their relative strengths (an interesting classification was proposed by Naor [17]. It is also important to analyse the security of cryptographic protocols not only from an theoretical point of view (e.g. adversary interacts with parties in a black-box mode) but in more realistic models wherein an opponent may physically attack an honest peer (e.g. tampering with devices [18]).

In any case, the publication of protocol specifications in specialised literature and conferences is of fundamental importance since the peer review process and the subsequent period of public scrutiny can increase the confidence in the security of the protocol either because vulnerabilities are not discovered or if there are proposals to fix them will be immediately published.

A topic for future research is the development of a concrete implementation of the protocols discussed in this paper to evaluate real world security issues and efficiency. In particular, we plan to develop an implementation with the Java Cryptography Architecture framework [19] which uses a "provider"-based approach and contains a set of APIs for various purposes (e.g. encryption, key generation and management, secure random number generation, certificate validation, etc).

REFERENCES

- [1] Abdalla M., Fouque P.A., Pointcheval D. 2005 *Password-based authenticated key exchange in the three-party setting* Proceedings of the PKC05, LNCS, 3386: 65-84
- [2] Tan A. 2012. *A Communication and computation-efficient three party authenticated key agreement protocol* Security Comm. Networks DOI: 10.1002/sec.622
- [3] Tan Z., 2010. *An enhanced three-party authentication key exchange protocol for mobile commerce environments* J.Comm., DOI:10.4304/jcm.5.5.436-443
- [4] Yang J.H., Chang C.C., 2009. *An efficient three-party authenticated key exchange protocol using elliptic curve cryptography for mobile-commerce environments* J. Syst. Software, DOI:10.1016/j.jss.2009.03.075
- [5] Chou Chih-ho, Tsai Kuo-yu, Wu Tzong-chen, Yeh Kuo-hu. 2013. *Efficient and secure three-party authenticated key exchange protocol for mobile environments*, 14(5):347-355
- [6] Bellare M. and Rogaway P. 1993 *Entity Authentication and Key Distribution* Advances in Cryptology-CRYPTO 93, pp. 110-125, Springer-Verlag.
- [7] Canetti R. and Krawczyk H. 2001 *Analysis of key exchange protocols and their use for building secure channels* Proc. of Eurocrypt'01, LNCS 2045, pp. 453-474.
- [8] Hankerson D., Menezes A.J. and Vanstone S.A. 2004 *Guide to Elliptic Curve Cryptography* Springer Professional Computing, New York
- [9] Matsumoto T., Takashima Y, and Imai H. 1986 *On seeking smart public-key distribution systems* Transactions of IEICE, VolE69:99-106
- [10] Lamacchia B., Lauter B. and Mityagin A. 2007 *Stronger Security of Authenticated Key Exchange* LNCS 4784, pp. 1-16.
- [11] Mohammad Z. and Chi-Chun Lo 2009 *Vulnerability of an improved Elliptic Curve Diffie-Hellmann Key Agreement and its Enhancement* Proc. of EBISS'09, pp. 1-5
- [12] Strangio M.A. 2006 *On the Resilience of Key Agreement Protocols to Key Compromise Impersonation* Cryptology ePrint Archive, <http://eprint.iacr.org/2006/252.pdf>.
- [13] Wang S, Cao Z, Strangio M.A and Wang B 2009 *Cryptanalysis and Improvement of an Elliptic Curve Diffie-Hellmann Key Agreement Protocol* IEEE Communication Letters, Vol. 12, No. 2, pp. 149-151.
- [14] Choo K.R, Boyd C, Hitchcock Y 2005 *Examining Indistinguishability-Based Proof Models for Key Establishment Protocols*, Advances in Cryptology-ASIACRYPT, Springer Berlin/Heidelberg, pp. 585-604
- [15] Boyd C, Mathuria A 2003 *Protocols for Authentication and Key Establishment*, Springer Berlin/Heidelberg
- [16] ECRYPT 2013 *Final Report on Main Computational Assumptions in Cryptography*, ECRYPT II, <http://www.ecrypt.eu.org/documents/D.MAYA.6.pdf>
- [17] Naor M. 2003 *On Cryptographic Assumptions and Challenges*, Advances in Cryptology, Lecture Notes in Computer Science Volume 2729, pp 96-109
- [18] Anderson R, 2001 *Security Engineering: A guide to Building Dependable Distributed Systems*, Wiley
- [19] Oracle, 2013 *Java Cryptography Architecture*, <http://docs.oracle.com/javase/7/docs/technotes/guides/security/crypto/CryptoSpec.html>
- [20] Bellare M., Rogaway P. 1995 *Provably secure session key distribution: the three party case*, Proceedings of the ACM Symposium on the Theory of Computing (STOC95), pp. 5766