On the Feasibility of Key Compromise Impersonation Attacks Against the Elliptic Curve Version of the MTI/C0 Key Agreement Protocol

Maurizio Adriano Strangio

***Abstract*—In a recent conference (ICSSSM 2010), Eun-Jun Yooni and Kee-Young proposed the two-pass elliptic curve Diffie-Hellman key agreement protocol EECKE-1N that according to the authors is more efficient that ECKE-1N. In this paper we point out that the protocol proposed by the aforementioned authors is in fact the elliptic curve version of the MTI/C0 family of key agreement protocols which suffer from a well-known vulnerability against a particular type of key-compromise impersonation attack.**

***Index Terms*—Key agreement protocols, elliptic curves, key-compromise impersonation.**

I. Introduction

Since the Diffie-Hellman (DH) key exchange scheme was published in 1976 [1], to address the problem of communicating on insecure networks, a large number of key agreement protocols based on this paradigm have been proposed by many researchers (see for example [2]-[10] and Section 12.6 of [11]).

The output of a DH key exchange protocol is a shared cryptographic key which two parties may use to setup a private and authenticated communication channel. To establish the key, a variety of cryptographic primitives are employed with entity authentication achieved via public key cryptography.

In a recent conference (ICSSSM 2010), Eun-Jun Yooni and Kee-Young proposed the two-pass elliptic curve Diffie-Hellman key agreement protocol EECKE-1N [12] which according to the authors has the same security properties of ECKE-1N [13] but is more efficient.

In this paper, we point out that the EECKE-1N protocol is in fact the specification of the MTI/C0 family of protocols on elliptic curves; it is well-known that the MTI/C0 protocol is vulnerable to a particular type of key compromise impersonation (KCI) attack (that we describe in Section 3).

II. Review of Protocol EECKE-1N

To make this work self-contained we first recall some important mathematical background and then briefly review the EECKE-1N protocol specification (Fig. 1).

Domain parameters are defined by the 8-tuple: $\Phi_{EC} = (q,$ FR, $S, a, b, P, n, h)$

Manuscript received May 13, 2012; revised June 14, 2012.

M. A. Strangio is with the Department of Mathematics, University of Rome "Roma Tre" (e-mail: strangio@mat.uniroma3.it).

where

- q is the underlying field order;
- FR (field representation) is an indication of the method used to represent field elements in F_q;
- the *seed S* is for randomly generated elliptic curves;
- the *coefficients* $a, b \in F_q$ define the equation of the elliptic curve $E(F_q)$ over F_q;
- the *base point* $P=(P.x, P.y)$ of large prime order in $E(F_q)$;
- the prime order n of P and the cofactor $h=\#E(F_q)/n$ (where $\#E(F_q)$ denotes the number of points in the curve $E(F_q)$).

The parameters Φ_{EC} should be appropriately chosen so that no efficient algorithms exist that solves the Discrete Logarithm Problem (DLP) or the Computational Diffie-Hellman Problem (CDHP) in the subgroup $<P>$. The point P_∞ denotes the identity point in $<P>$. The domain parameters must also undergo a validation process proving the elliptic curve has the claimed security attributes [14].

Capital letters A,B are used to denote principals; their private-public key pairs are, respectively, (w_A,W_A) and (w_B,W_B) with w_A randomly chosen in [1, n-1] and $W_A=w_AP$. We assume that digital certificates (denoted by $cert_A$, $cert_B$ respectively) are issued by mutually trusted Certification Authorities (CA). The map $H: F_q \rightarrow \{0,1\}^l$ represents a hash function ($l > 128$).

A: (w_A,W_A), B: (w_B,W_B)	
A:	$r_A \in_R [1,n\text{-}1]$
	$Q_A = r_A W_B$
$A \rightarrow B$	Q_A
B:	$r_B \in_R [1,n\text{-}1]$
	$Q_B = r_B W_A$
$B \rightarrow A$	Q_B
A:	$T_A=h\, w_A^{-1}\, r_A\, Q_B$
	$sk=H(T_A.x, Q_A.x, Q_B.x, id_A, id_B)$
B:	$T_B=h\, w_B^{-1}\, r_B\, Q_A$
	$sk=H(T_B.x, Q_A.x, Q_B.x, id_A, id_B)$

Fig. 1. Protocol EECKE-1N.

The actions performed by each peer are the following:

1) A chooses a random nonce $r_A \in_R [1,n\text{-}1]$ (the symbol $\in_R$ denotes the selection of an element at random in a set) and computes $Q_A=r_AW_B$;
2) If $Q_A=P_\infty$ return to step 1. Otherwise, A sends Q_A to B to initiate the key exchange protocol;

3) B invokes a procedure to perform public-key validation of Q_A (e.g. to verify that Q_A is actually a point in the group $E(F_q)$ and aborts the protocol run if the validation fails. Otherwise, B chooses $r_B \in_R [1,n-1]$, computes $Q_B=r_B W_A$ and sends Q_B to A as the response message (B also repeats the last step if $Q_B=P_\infty$);
4) A performs public-key validation of Q_B and aborts the protocol run if the validation fails;
5) A and B compute, respectively, the points T_A and T_B;
6) Both A and B terminate holding the session key *sk*.

Correctness of the protocol follows by observing that, for honest parties A and B, at the end of a protocol run, if the equality $T_A=T_B=hr_A r_B P$ holds then both peers will compute the same session key *sk*.

It is a simple matter to verify that the above protocol scheme is equivalent to the original specification of the MTI/C0 [6] protocol in a cyclic group G of prime order p (with g generator of G), where scalar multiplication correspond to exponentiations; indeed, in this case the keying material would be computed as follows

$$Q_A^{r_A/w_A}=(g^{w_A/r_B})^{r_A/w_A}=(g^{r_A/r_B})=(g^{w_B/r_A})^{r_B/w_B}=Q_B^{r_B/w_B}.$$

III. A KCI Attack Against Protocol EECKE-1N

In this section we show that protocol EECKE-1N is vulnerable to KCI attacks. Recall that in a KCI attack the adversary E, having learned A's long-term private, attempts to establish a valid session key with A by masquerading as another legitimate principal (say B).

It is instructive to discuss the implications deriving from the exposure of a private key [15], for example, the adversary can 1) impersonate A to another principal; 2) impersonate another principal (say B) in a protocol run with A (KCI attack); and 3) attempt to learn the session keys established in earlier runs of the protocol between honest parties.

In case 1) the adversary can convince B to accept a message that was not truly sent by A while, in case 2) she can send messages to A as if they originated from any other principal. In practice, the adversary could impersonate a banking system and cause party A to accept a predetermined session key and then obtain personal data (e.g. a credit card number) over the secure communication link. In case 3) the adversary may be able to decrypt the data exchanged by A and B in previous runs of the protocol (provided the communication transcripts are known).

In Fig. 2 we provide a detailed description of the KCI attack against protocol EECKE-1N (E(B) denotes that E is impersonating B):

1) E(B) (posing as B) prompts A to initiate a session with B;
2) A chooses $r_A \in_R [1,n-1]$, computes $Q_A=r_A W_B$ and sends Q_A to B (the intended recipient);
3) E(B) intercepts Q_A and relays it to B without modifications. B's response (Q_B) is deleted from the network and replaced by $Q_{E(B)}=r_{E(B)} w_A W_B$ for some random $r_{E(B)} \in_R [1,n-1]$. Message $Q_{E(B)}$ is delivered to A;
4) A and E(B) compute, respectively, the points $T_A=T_{E(B)}$. Both A and E(B) terminate holding the session key *sk* (see below) and therefore the attack is successful.

It is a trivial matter to prove that $T_A=T_{E(B)}$:

$$\begin{aligned} T_A &= hw_A^{-1} r_A Q_{E(B)} \\ &= hw_A^{-1} r_A r_{E(B)} w_A W_B \\ &= hr_A r_{E(B)} W_B \\ &= hr_{E(B)} Q_A \\ &= T_{E(B)}. \end{aligned}$$

Therefore, whenever A initiates a secure communication with some known party, E can always intercept the first message Q_A and subsequently impersonate the peer to A, until the compromise is detected and the long-term key is revoked.

A: (w_A,W_A), B: (w_B,W_B)	
A:	$r_A \in_R [1,n-1]$
	$Q_A = r_A W_B$
$A \rightarrow B$	Q_A
$E(B)$:	$r_{E(B)} \in_R [1,n-1]$
	$Q_{E(B)} = r_{E(B)} w_A W_B$
$E(B) \rightarrow A$	$Q_{E(B)}$
A:	$T_A=h\, w_A^{-1}\, r_A\, Q_{E(B)}$
	sk=H($T_A.x$, $Q_A.x$, $Q_{E(B)}.x$, id_A, id_B)
B:	$T_{E(B)}=h\, r_{E(B)}\, Q_A$
	sk=H($T_B.x$, $Q_A.x$, $Q_{E(B)}.x$, id_A, id_B)

Fig. 2. KCI attack on protocol EECKE-1N

IV. On the Security of the EECKE-1N Protocol

In this section we qualitatively evaluate the security of the EECKE-1N (MTI/C0) protocol under the complexity-theoretic model of authenticated key exchange (AKE) described in [9]. In this model (denoted eCK) the adversary is given the ability to ask of any party the following types of queries:

5) Ephemeral Key Reveal, to obtain all session specific information used by the party in a protocol run;
6) Long-Term Key Reveal, which returns all secret information (e.g. long-term private keys) and
7) Reveal, for exposing session keys. Any combination of queries allowing the adversary to trivially break an AKE protocol is not permitted.

Test sessions are classified as either “passive” or “active" depending on whether the adversary can (respectively) only observe the communication transcripts exchanged by two honest participants or also modify them. As the result, in passive sessions both parties will possibly complete matching sessions; on the other hand, active sessions are those where matching sessions are not necessarily established.

For passive sessions we allow the adversary to reveal any subset of the four types of secret keys (ephemeral and long-term secrets), held by the two parties, which does not contain both the ephemeral and long-term secret keys of one peer. In other words, simultaneous Ephemeral Key Reveal and Long-Term Key Reveal queries of one party are not

allowed in the eCK model since they would trivially allow the adversary to break any AKE protocol. When the adversary is allowed to forge communications between parties, i.e. for active sessions, she can only reveal the long-term secret key or the ephemeral secret key of the party which is executing the test session.

Technically, the protocol is broken in the eCK model if the test session (on which the adversary issues the test query) is clean and the guess query returns 1. A session sid initiated by A (session identifiers are defined as the concatenation of the identities of two peers and the information they exchange during a protocol run) is not clean if any of the following conditions hold:

a) A or B is totally controlled by the adversary;

b) the adversary issues a query Reveal(sid) or Reveal(sid*) if the later session exists (the session identifier sid* is relative to the peer B);

c) session sid* exists and the adversary issues queries Long-Term Key Reveal(A) and Ephemeral Key Reveal(sid) or Long-Term Key Reveal(B) and Ephemeral Key Reveal(sid*);

d) session sid* does not exist and the adversary issues query either Reveal(sid*) or both Long-Term Key Reveal(A) and Ephemeral Key Reveal(sid*).

The adversary E runs the AKE experiment by executing the algorithm described in Fig. 2. E chooses the session initiated by A as the test session and after sending message $Q_{E(B)}$ to A asks the query `Long-Term Key Reveal`(A); at some stage E queries the random oracle with the same arguments used by the party running the test session (i.e. $T_A.x$, $Q_A.x$, $Q_{E(B)}.x$, id_A, id_B). The attack succeeds because the adversary knows the value of T_A and the test session is clean since none of the aforementioned conditions are true.

REFERENCES

[1] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol 22, no. 6, pp. 644-654, 1976

[2] S. B. Wilson and A. J. Menezes, "Authenticated Diffie-Hellman key Agreement Protocols," in *Proc. Selected Areas in Cryptography*, pp. 339-361, 1999.

[3] R. Canetti and H. Krawczyk, "Analysis of key exchange protocols and their use for building secure channels," in *Proc. of Eurocrypt* 2001, pp. 453-474, 2001

[4] B. Song and K. Kim, "Two-pass authenticated key agreement protocol with key confirmation," in *Proc. of Indocrypt* 2000, pp. 237-249, 2000

[5] M. A. Strangio, "Efficient Diffie-Hellman two-party key agreement protocols based on elliptic curves," in *Proc. of ACM Symposium on Applied Computing*; pp. 324-331, 2005

[6] L. Law, A. J. Menezes, M. Qu, J. Solinas, and S. Vanstone, "An efficient protocol for authenticated key agreement," *Univ. of Waterloo*, pp. 98-05, 1998

[7] H. Krawczyk, "HMQV: A high performance secure Diffie-Hellman protocol," in *Proc. of Crypto* 2005*;* LNCS 3621, pp. 546-566, 2005

[8] T. Matsumoto, Y. Takashima, and H. Imai, "On seeking smart public-key distribution systems," *Trans. IEICE Jpn;* vol E69-E, no. 2, pp. 99-106, 1986

[9] B. Lamacchia, K. Lauter, and A. Mityagin, "Stronger security of authenticated key exchange," LNCS 4784, pp. 1-16, 2007

[10] R. Ankney, D. Johnson, and M. Matyas, "The unified model," Contribution *to X9F1,* 1995.

[11] A.J. Menezes, P. C. Van Oorcchot, and S. Vanstone, *Handbook of applied cryptography* Boca Raton: CRC Press, 1997

[12] Y. E. Jun and K. Young, "A new elliptic curve Diffie-Hellman two-party key agreement protocol," *Int.l Conference on Service System and Service Management (ICSSSM 2009),* pp. 1-4, 2010

[13] S. Wang, Z. Cao, and M.A. Strangio, and L. Wang, "Cryptanalysis and improvement of an elliptic curve Diffie-Hellman key agreement protocol," *IEEE comm. Letters;* vol. 12, no. 2, pp. 149-151, 2008

[14] D. Hankerson, A.J. Menezes, and S. A. Vanstone, *Guide to elliptic curve cryptography*, New York: Springer Professional Edition, 2004

[15] M. A. Strangio, "On the resilience of key agreement protocols to key compromise impersonation," *Cryptology ePrint Archive Report 2006/252*, 2006

Maurizio A. Strangio received his B.S. degree in computer science from the University of Bari, Italy in 1989. He also received his PhD (information security) from the University of "Tor Vergata", Rome, Italy in 2006. His research interests include cryptography, computer forensics, computer security and signal processing.