Cryptanalysis and Improvement of an Elliptic Curve Diffie-Hellman Key Agreement Protocol *

Shengbao Wang[1], Zhenfu Cao[1], Maurizio Adriano Strangio[2] and Lihua Wang[3]

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
{shengbao-wang,cao-zf}@cs.sjtu.edu.cn
[2] Department of Mathematics, University of Rome "Roma Tre", Italy
strangio@mat.uniroma3.it
[3] Information Security Research Center, National Institute of Information and Communications Technology, Japan
wlh@nict.go.jp

December 14, 2007

Abstract. In SAC'05, Strangio proposed protocol ECKE-1 as an efficient elliptic curve Diffie-Hellman two-party key agreement protocol using public key authentication. In this letter, we show that despite the author's claims protocol ECKE-1 is vulnerable to key-compromise impersonation attacks.

We also present an improved protocol — ECKE-1N, which can withstand such attacks. The improved protocol's performance is comparable to the well-known MQV protocol and maintains the same remarkable list of security properties.

Key Words. Key agreement, elliptic curve cryptography, Diffie–Hellman protocol, key-compromise impersonation, MQV.

1 Introduction

Since the Diffie-Hellman key exchange scheme was published [4], a large number of key agreement protocols have been proposed (see [2] and Section 12.6 of [12] for comprehensive surveys).

A secure (two-party) key agreement protocol should not allow a resource constrained adversary, eavesdropping or manipulating message flows in a finite number of protocol runs, to subvert any of the security goals (e.g. obtain information on the secret session key, engage in a successful protocol run while masquerading as a legitimate principal, etc). However, the design of secure and efficient key agreement protocols is notoriously far from being a simple task; there are so many details involved (including the complicated interactions with the environment) that the designer cannot establish beyond doubt that his protocol is infallible. This holds regardless of whether security proofs are supported by heuristic arguments or developed in formal models of distributed computing. In practice, the degree of confidence accompanying a protocol (as with many other cryptographic primitives) increases with time

* This is the full version of a letter to appear in IEEE Communications Letters. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

as the underlying algorithms (and assumptions) survive many years of public scrutiny without any significant flaws being discovered.

With protocols affording *implicit key authentication* (IKA) one party is ensured that no other party aside from its intended peer may learn the established secret key. Key agreement protocols that provides mutual IKA between the two parties are generally known as *authenticated key agreement* (AK) protocols [2].

In SAC'05, Strangio [13] proposed an efficient two-pass elliptic curve Diffie-Hellman key agreement protocol (ECKE-1) that makes use of public key authentication. This protocol belongs to the class of Diffie-Hellman based key exchange schemes [4] affording *implicit key authentication* (IKA), i.e. both parties are ensured that no other principals aside from their intended peers may learn the established secret key. The author claimed that protocol ECKE-1 enjoys important security attributes such as known-key security (K-KS), forward secrecy (FS), unknown key-share resilience (UK-SR), key control (KC), and *key-compromise impersonation resilience* (K-CIR).

In this letter we show that protocol ECKE-1 is vulnerable to key-compromise impersonation attacks and present an improved protocol ECKE-1N which is key-compromise impersonation resilient. Notably, protocol ECKE-1N achieves performance figures and security properties that are comparable to those of the mainstream MQV protocol.

The rest of the paper is organized as follows. We first review protocol ECKE-1 [13] in Section 2. In Section 3, we provide the details of the key-compromise impersonation attack against protocol ECKE-1. In Section 4 we present protocol ECKE-1N while Section 5 contains our concluding remarks.

2 Review of Protocol ECKE-1

We briefly review protocol ECKE-1 (Figure 1, [13]). Domain parameters are defined by the 8-tuple

$$\Phi_{EC} = (q, FR, S, a, b, P, n, h)$$

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

The parameters Φ_{EC} should be appropriately chosen so that no efficient algorithms exists that solve the Discrete Logarithm Problem (DLP) or the Computational Diffie-Hellman Problem (CDHP) in the subgroup $\langle P \rangle$. The point P_∞ denotes the identity point in $\langle P \rangle$. The domain parameters must also undergo a validation process proving the elliptic curve has the claimed security attributes [5].

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 $w_A \in_R [1, n-1]$ and $W_A = w_A P$. We assume that digital certificates (denoted by $cert_A, cert_B$ respectively) are issued by mutually trusted Certification Authorities (CA). The maps $\mathcal{F}_1, \mathcal{F}_2 : \{0,1\}^* \rightarrow \mathbb{F}_q$ represent two independent hash functions and $\mathcal{G} : \mathbb{F}_q \rightarrow \{0,1\}^\ell$ a key derivation function ($\ell \geq 128$).

1. A picks a random $r_A \in [1, n-1]$ and computes $e_A = \mathcal{F}_1(r_A, w_A, id_A)$. Analogously, B picks a random r_B and computes $e_B = \mathcal{F}_1(r_B, w_B, id_B)$;
2. A computes $Q_A = (r_A + e_A w_A)P$. Symmetrically, B computes $Q_B = (r_B + e_B w_B)P$;
3. If $Q_A = P_\infty$ (resp. $Q_B = P_\infty$), A (resp. B) returns to step 2. Otherwise, A initiates a protocol run with B by sending Q_A to B;

$$
\begin{array}{ll}
\hline
\multicolumn{2}{l}{A(w_A, W_A), B(w_B, W_B)} \\
\hline
 & A : r_A \in_R [1, n-1] \\
 & \quad e_A = \mathcal{F}_1(r_A, w_A, id_A) \\
 & \quad Q_A = (r_A + e_A w_A)P \\
A \to B\text{:} & Q_A \\
 & B : r_B \in_R [1, n-1] \\
 & \quad e_B = \mathcal{F}_1(r_B, w_B, id_B) \\
 & \quad Q_B = (r_B + e_B w_B)P \\
B \to A\text{:} & Q_B \\
 & A : d_A{=}w_A \mathcal{F}_2(Q_A.x, Q_B.x, id_A, id_B) \\
 & \quad T_A = h((r_A + e_A w_A)Q_B + d_A W_B) \\
 & \quad sk = \mathcal{G}(T_A.x) \\
 & B : d_B{=}w_B \mathcal{F}_2(Q_A.x, Q_B.x, id_A, id_B) \\
 & \quad T_B = h((r_B + e_B w_B)Q_A + d_B W_A) \\
 & \quad sk = \mathcal{G}(T_B.x) \\
\hline
\end{array}
$$

Fig. 1. Protocol ECKE-1

4. 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 $E(\mathbb{F}_q)$) and aborts the protocol run if the validation fails. Otherwise, B sends Q_B to A as the response message;
5. A performs public-key validation of Q_B and aborts the protocol run if the validation fails;
6. A and B compute, respectively, the points T_A and T_B;
7. Both A and B terminate holding the session key sk.

Correctness of the protocol follows from the equality $T_A = T_B$; in this case honest parties A and B will both compute the same session key from the elliptic curve point[4] $h(r_A r_B + r_B e_A w_A + r_A e_B w_B + e_A e_B w_A w_B + d w_A w_B)P$ where $d = \mathcal{F}_2(Q_A.x, Q_B.x, id_A, id_B)$.

The scalar multiplication using the cofactor h prevents the small-subgroup attack [9].

3 A K-CI Attack on Protocol ECKE-1

In this section we show that protocol ECKE-1, contrary to the author's claims [13], suffers from a vulnerability that exposes it to key-compromise attacks.

Suppose the long-term private key of a principal A is compromised by the adversary E. Obviously, E is now able to impersonate the corrupted party to any other party. However, it is also desirable that knowledge of the private key does not enable the adversary to impersonate other entities to the corrupted party. Accordingly, a *key-compromise impersonation attack* is an attack whereby E, with A's long-term private key at hand, attempts to establish a valid session key with A by masquerading as another legitimate principal (say B). Note that key-compromise impersonation attack represents a serious threat since a party may not be (immediately) aware that her private key was compromised.

A detailed description of the K-CI attack against protocol ECKE-1 is outlined below (see also Figure 2 — $E(B)$ denotes that E is impersonating B):

1. $E(B)$ (posing as B) "prompts" A to initiate a session with B;
2. A chooses a random $r_A \in [1, n-1]$, computes $e_A = \mathcal{F}_1(r_A, w_A, id_A)$ and sends $Q_A = (r_A + e_A w_A)P$ to B (the intended recipient);

[4] Notice that there is a typo in [13], where the author mistakenly writes this term as $h(r_A r_B + r_B e_A w_B + r_A e_B w_A + e_A e_B w_A w_B + d_A d_B w_A w_B)P$.

$A(w_A, W_A), B(w_B, W_B)$	
$A :$	$r_A \in_R [1, n-1]$
	$e_A = \mathcal{F}_1(r_A, w_A, id_A)$
	$Q_A = (r_A + e_A w_A)P$
$A \to B$:	Q_A
$E(B) :$	$r_{E(B)} \in_R [1, n-1]$
	$Q_{E(B)} = r_{E(B)}P$
$E(B) \to A$:	$Q_{E(B)}$
$A :$	$d_A{=}w_A\mathcal{F}_2(Q_A.x, Q_{E(B)}.x, id_A, id_B)$
	$T_A = h((r_A + e_A w_A)Q_{E(B)} + d_A W_B)$
	$sk = \mathcal{G}(T_A.x)$
$E(B) :$	$d_{E(B)}{=}w_A\mathcal{F}_2(Q_A.x, Q_{E(B)}.x, id_A, id_B)$
	$T_{E(B)} = h(r_{E(B)}Q_A + d_{E(B)}W_B)$
	$sk = \mathcal{G}(T_{E(B)}.x)$

Fig. 2. K-CI attack on protocol ECKE-1.

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 $Q_{E(B)} = r_{E(B)}P$ for some random $r_E(B) \in [1, n-1]$. 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.

We now prove that $T_A = T_{E(B)}$ as follows (obviously, we have $d_{E(B)} = d_A$):

$$\begin{aligned}
T_A &= h((r_A + e_A w_A)Q_{E(B)} + d_A W_B) \\
&= h((r_A + e_A w_A)r_{E(B)}P + d_A W_B) \\
&= h(r_{E(B)}(r_A + e_A w_A)P + d_{E(B)}W_B) \\
&= h(r_{E(B)}Q_A + d_{E(B)}W_B) \\
&= T_{E(B)}.
\end{aligned}$$

Therefore, when A wants to initiate a secure communication with any specific entity, E can always intercept the first protocol message Q_A and subsequently impersonate the specific entity to A, until the compromise is detected and the long-term key is revoked.

4 An Improved Protocol — ECKE-1N

In this section we present protocol ECKE-1N which is key-compromise impersonation resilient. The specification of protocol ECKE-1N is shown in Figure 3.

As in [6], A and B must make sure that $Q_B \neq P_\infty$, $Q_A \neq P_\infty$, respectively.

Correctness of the protocol immediately derives from the equality $T_A = T_B = h(r_A + e_A)(r_B + e_B)P$. The map $\mathcal{H} : \{0,1\}^* \to \mathbb{F}_{|q|/2}$ is a collision resistant hash functions which outputs $|q|/2$ bits. As a consequence, the on-line computational effort for each principal is mostly due to the 2.5 scalar multiplications, one field multiplication and one field inversion.

4.1 Security Arguments

We first show that protocol ECKE-1N is resilient to K-CI attacks. Suppose the adversary E has learned the long term private key w_A of principal A; she is now able to set up a man-in-the-middle attack during a run of the protocol between A and B. The attack should work as

$A(w_A, W_A), B(w_B, W_B)$
$A : r_A \in_R [1, n-1]$
$Q_A = r_A W_B$
$e_A = \mathcal{H}(Q_A, id_B, id_A)$
$A \to B$: Q_A
$B : r_B \in_R [1, n-1]$
$Q_B = r_B W_A$
$e_B = \mathcal{H}(Q_B, id_A, id_B)$
$B \to A$: Q_B
$A : e_B = \mathcal{H}(Q_B, id_A, id_B)$
$T_A = hw_A^{-1}(r_A + e_A)(Q_B + e_B W_A)$
$sk = \mathcal{G}(T_A.x)$
$B : e_A = \mathcal{H}(Q_A, id_B, id_A)$
$T_B = hw_B^{-1}(r_B + e_B)(Q_A + e_A W_B)$
$sk = \mathcal{G}(T_B.x)$

Fig. 3. Protocol ECKE-1N

follows. E lets message Q_A reach its intended destination (B) but replaces B's response Q_B with X. On receipt of X, A computes the elliptic curve point $T_A = hw_A^{-1}(r_A + e_A)(X + e_B W_A)$. Algorithm E receives in input the data w_A, Q_A, Q_B, W_A, W_B and must output the value T_A computed by A. A straightforward strategy for E is to compute r_A; however, extracting r_A from Q_A is unfeasible for the adversary since by our assumptions the Discrete Logarithm Problem (DLP) is intractable in the underlying elliptic curve group.

Now, the question is whether E is able to choose a suitable message X, in order to cancel the terms that depend on r_A from T_A, by exploiting the algebraic properties of the group (similarly to the attacks of [13]). In fact, it appears that the term $e_B W_A$ can be eliminated by choosing $X = r_E W_B - e_B W_A$ since we would have $T_A = hw_A^{-1}(r_E Q_A + e_A r_E W_B)$. However, E is unable to determine such an X since she must solve the non-linear recursive equation $X = r_E W_B - \mathcal{H}(X, id_B, id_A) W_A$.

The protocol also enjoys other important security attributes. Forward secrecy is achieved by means of the term $r_A r_B P$ (common factor of T_A, T_B) and holds due to the intractability of the Computational Diffie-Hellman Problem (CDHP). Note that here we refer to the weaker form of forward secrecy that involves a passive adversary (who knows the long-term private keys of both peers) eavesdropping on a session of the protocol and then attempting to expose the key [6].

The inclusion of both identities (id_A, id_B) in the terms e_A, e_B can preclude UK-S attacks since they are involved in the calculation of the session key and therefore the replacement of a certificate (e.g. the public keys of A, B registered with a different identity) would not allow the communication to take place (the parties would accept different keys).

The conjectured security attributes of several one-round elliptic curve Diffie-Hellmann key agreement protocols that use public key authentication are summarised in Table 1.

The first column indicate whether the protocols enjoy implicit key authentication (IKA). Column two shows that all protocols satisfy the basic key independence (K-KS) security requirement while only protocol MTI/A0 does not provide forward secrecy (column three). Column four reveals that ECKE-1N enjoys K-CI resilience together with the MQV, HMQV and MTI/A0 protocols. Finally, columns five (UK-SR) and six (KC) show that all listed protocols enjoy the unknown key share resilience and key control (the abbreviation "init" refers to the initiator party) security attributes respectively.

Table 1. Conjectured security attributes for one-round key agreement protocols

↓*Prot./Sec.Attrib.*→	*IKA*	*K-KS*	*FS*	*K-CIR*	*UK-SR*	*KC*
MTI/A0[11]	yes	yes	no	yes	yes	init
UM[1]	yes	yes	yes	no	yes	init
MQV[9]	yes	yes	yes	yes	yes	init
HMQV[6]	yes	yes	yes	yes	yes	init
LLK[7]	yes	yes	yes	no	yes	init
SK[14]	yes	yes	yes	no	yes	init
ECKE-1[13]	yes	yes	yes	no	yes	init
ECKE-1N	yes	yes	yes	yes	yes	init

Additionally, we note that by adopting the elegant idea from [8], namely hashing ephemeral and long-term private keys, our protocol provides resilience to the leakage of ephemeral private keys (see [10] for more details).

4.2 Computational Efficiency

Table 2. Performance comparison of one-round key agreement protocols

↓*Prot./Computation*→	*Point Mult.*	*Field Mult.*	*Hash*	*Field inversion*
MTI/A0	3	0	0	0
UM	3	0	0	0
MQV	2.5	1	0	0
HMQV	2.5	1	2	0
LLK	2	1	0	1
SK	3	1	0	0
ECKE-1	3	2	2	0
ECKE-1N	2.5	1	2	1

The computational effort required by each principal in the above protocols is reported in Table 2. Column one counts the number of exponentiations while column two shows the number of field multiplications. Hash function calculations are enumerated in column three (key derivation functions are omitted since they apply to all protocols — note also that some hash computations can be done off-line). Finally, column four displays the number of field inversions.

5 Conclusions

Key agreement protocols play a central role for achieving secure communications in hostile networks; however, protocol design is extremely error-prone due to the inherent complexity of the problem.

In this letter we have shown that protocol ECKE-1 [13] is insecure against key-compromise impersonation attacks. We have also presented an improved protocol ECKE-1N that can withstand such attacks and achieves overall performance and security comparable to the well-known standardized MQV protocol.

Work is currently in progress to formally prove the security of the protocol in a model of distributed computing (e.g. [3,8]).

Acknowledgments

This work was supported in part by the National High Technology Development Program of China under Grant No. 2006AA01Z424 and the National Natural Science Foundation of China under Grant Nos. 60673079, 60572155 and 60773086.

References

1. R. Ankney, D. Johnson and M. Matyas, "The Unified Model," Contribution to X9F1, 1995.
2. S. Blake-Wilson and A. Menezes, "Authenticated Diffie-Hellman key agreement protocols," in *Proc. Selected Areas in Cryptography 1998*, LNCS 1556, 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'01*,LNCS 2045, pp. 453-474, 2001.
4. W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory* vol.22, no.6, pp. 644-654, 1976.
5. D. Hankerson, A.J. Menezes and S.A. Vanstone, "Guide to elliptic curve cryptography," Springer Professional Computing, New York, 2004.
6. H. Krawczyk, "HMQV: A high performance secure Diffie-Hellman protocol," in *Proc. of Crypto'05*, LNCS 3621, pp. 546-566, 2005.
7. C. Lee and J. Lim and J. Kim, "An efficient and secure key agreement," IEEE p1363a draft, 1998.
8. B. LaMacchia, K. Lauter and A. Mityagin, "Stronger security of authenticated key exchange ," in *Proc. of ProvSec'07*, LNCS 4784, pp. 1-16, 2007.
9. L. Law, A. Menezes, M. Qu, J. Solinas, and S. Vanstone, "An efficient protocol for authenticated key agreement," Dept. C & Q, Univ. of Waterloo, CORR 98-05, 1998.
10. B. Ustaoglu, "Obtaining a secure and efficient key agreement protocol from (H)MQV and NAXOS," Cryptology ePrint Archive, Report 2007/123, 2007.
11. 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.
12. A. Menezes, P.C. van Oorschot, and S. Vanstone, Handbook of Applied Cryptography. CRC Press, Boca Raton, 1997.
13. M. A. Strangio, "Efficient Diffie-Hellmann two-party key agreement protocols based on elliptic curves," in *Proc.* 20^{th} *ACM Symposium on Applied Computing (SAC)*, pp. 324-331, 2005.
14. B. Song and K. Kim, "Two-Pass authenticated key agreement protocol with key confirmation," in *Proc. of Indocrypt'00*, LNCS 1977, pp. 237-249, 2000.