Proceedings, International Conf. on Security and Cryptography (SECRYPT) 2007.
Part of the International Joint Conference on E-business and Telecommunications (ICETE).

ON THE KEY-COMPROMISE IMPERSONATION VULNERABILITY OF ONE-PASS KEY ESTABLISHMENT PROTOCOLS

K. Chalkias, F. Mpaldimtsi, D. Hristu-Varsakelis and G. Stephanides
Computational Systems and Software Engineering Laboratory
Department of Applied Informatics
University of Macedonia
156 Egnatia St.
Thessaloniki, Greece
{*chalkias, foteini*}*@java.uom.gr,* {*dcv, steph*}*@uom.gr*

Keywords: Two-party key establishment, one-pass protocols, key-compromise impersonation, one-way channel.

Abstract: Key establishment protocols are among the most important security mechanisms via which two or more parties can generate a common session key to in order to encrypt their communications over an otherwise insecure network. This paper is concerned with the vulnerability of one-pass two-party key establishment protocols to key-compromise impersonation (K-CI) attacks. The latter may occur once an adversary has obtained the long-term private key of an honest party, and represents a serious — but often underestimated — threat. This is because an entity may not be aware that her computer has been compromised and her private key is exposed, and because a successful impersonation attack may result in far greater harm than the reading of past and future conversations. Our aim is to describe two main classes of K-CI attacks that can be mounted against all of the best-known one-pass protocols, including MQV and HMQV. We show that one of the attacks described can be somewhat avoided (though not completely eliminated) through the combined use of digital signatures and time-stamps; however, there still remains a class of K-CI threats for which there is no obvious solution.

1 INTRODUCTION

In order for two parties to communicate securely over an unreliable public network, they must be able to authenticate one another and agree on a secret encryption key. To achieve this, key establishment protocols are used at the start of a communication session in order to verify the parties' identities and establish a common session key. There are two basic categories of protocols (Blake-Wilson and Menezes, 1998). The first includes so-called *key transport* protocols, in which the session key is created by one entity and is securely transmitted to the other. A second category includes *key agreement* protocols, where information from both entities is used to derive the shared secret key. A protocol is said to be *symmetric* if both entities a-priori possess some common secret data, and *asymmetric* if the two entities share only authenticated public information such as a public key with a digital certificate.

Since the introduction of the Diffie-Hellman key exchange (Diffie and Hellman, 1976), there has been a large number of key establishment protocols proposed, including recent one-round (Jeong et al., 2004; Law et al., 1998), two-round (Bird et al., 1991; Lu et al., 2005) and three-round approaches (Blake-Wilson and Menezes, 1998; Boyd et al., 2004; Kwon, 2001). Some of the disadvantages of these protocols are their high computational and communication cost which, combined with their round complexity, make them unsuitable for use in one-way communication channels. At the same time, there are a variety of applications that require low-cost one-way channel communication. Some of the best-known examples include e-mail and SMS, where the receiver cannot immediately reply, store-and-forward applications (e.g., printers) where messages are sent to resources which need not reply at all, and secure key exchange in mobile environments where low communication cost is critical.

To satisfy these requirements, efficient scalable one-pass key establishment protocols have been developed recently (Law et al., 1998; Krawczyk, 2005). In those schemes, only one of the parties transmits information in order to create the session key (but does not transmit the key itself). This means that one-pass

approaches lie somewhere between the key transport and key agreement categories[1]. Furthermore, most, if not all, have been derived from modifications of pre-existing *x*-round protocols.

Almost all one-pass approaches belong to the category of authenticated key establishment (AK) protocols, because they provide *implicit key authentication (IKA)*, meaning that the two (uncorrupted) parties using the protocol are assured that no one else can possibly learn the value of their session key. On the other hand, one-pass protocols cannot achieve *known key security* (K-KS) because an adversary can simply replay a previous protocol run that he has managed to record; nor can they provide *perfect forward secrecy* (PFS) because there can be no protocol for implicit authentication that achieves PFS with two or fewer messages (Krawczyk, 2005). Finally, one-pass approaches are prone to *key-compromise impersonation* (K-CI) attacks, in a number of ways which will be discussed shortly.

Arguably, protocol designers are often more concerned with PFS, and seem to ignore K-CI (Strangio, 2006). However, K-CI can potentially have more serious consequences: besides reading past or future conversations, an attacker would also be able to elicit additional information that may never have been communicated otherwise, by masquerading as a different honest principal. Because of this, it is our opinion that more emphasis should be given on a protocol being *K-CI*–resistant. In this paper, we discuss and demonstrate a series of impersonation attacks that affect one-pass key establishment protocols, after a key-compromise has occurred. We also examine the use of time-stamps and standard digital signatures for the purpose of withstanding certain types of K-CI attacks. To the best of our knowledge, this is the first detailed study of such attacks on one-pass key establishment protocols.

The remainder of this paper is organized as follows: In Section 2 we fix notation and review some required definitions. Section 3 describes some of the best known one-pass two-party key establishment protocols. Section 4 discusses the K-CI vulnerability vis-a-vis a series of important and widely-used applications, and describes two basic types of K-CI attacks and possible responses.

[1]For this reason, it seems more appropriate to speak of one-pass *key establishment* as opposed to *key agreement*, as is done in most of the literature.

2 NOTATION AND PRIMITIVES

The protocols described in the next Section can be defined over any finite commutative group $\mathbb{G}$ of order n, that comes equipped with a difficult discrete logarithm problem. Throughout this paper we consider asymmetric protocols based on elliptic curve cryptosystems ($\mathbb{G}$ will be the group of points on an elliptic curve), and we will use additive representation for group operations (Kaliski, 2001). We will let P denote a generator of $\mathbb{G}$, and will assume that $\mathbb{G}$, P and n are fixed and known in advance to the parties. We will write cP to denote scalar multiplication, where $c \in \mathbb{Z}_n^*$.

The security of the protocols discussed next is linked to the following problems, whose solution is assumed to be difficult to compute in polynomial time:

Definition 1 Discrete Log Problem (DLP)
Given $P, Q \in \mathbb{G}$, find an integer $a \in Z_n^*$ such that $Q = aP \in \mathbb{G}$.

Definition 2 Computational Diffie-Hellman Problem (CDHP)
Given $P, aP, bP \in \mathbb{G}$, for some unknown $a, b \in Z_n^*$, find $abP \in \mathbb{G}$.

In the following we will apply hash functions and signature schemes to lists of several arguments. In such cases, we are going to write function arguments separated by commas, e.g., example $H(X,Y,Z)$. By doing so, we assume that we have a collision-free encoding which maps lists of arguments to binary strings, and that the parties' identities are arbitrary binary strings.

An entity, say $\hat{A}$, participating in a protocol is assigned a static *key pair* (a,A) which consists of a *public* and a *private key*. Public keys (denoted by upper case letters) are elements of $\mathbb{G}$, while private keys (denoted by the corresponding lower case letters) are elements of $\mathbb{Z}_n^*$. For example, the private key a may correspond to the public key $A = aP$.

Public keys are registered with a trusted directory, called the certificate authority (CA). The CA registers arbitrary keys with the restriction that no party can have more than one registered public key. We assume that all honest parties have a priori generated their public keys and have registered them with the CA, so that they can be known to and verified by other parties during protocol execution.

Table 1 lists the notation used throughout the paper.

Table 1: Notation.

$\hat{A}$, $\hat{B}$	identities of two communicating parties
P	generator of the group $\mathbb{G}$
n	prime order of $\mathbb{G}$
a, b	static private keys of $\hat{A}$ and $\hat{B}$, $a, b \in \mathbb{Z}_n^*$
A, B	static public keys of $\hat{A}$ and $\hat{B}$, $A = aP, B = bP$
r	ephemeral private key
R	ephemeral public key, $R = rP$
sk_i	session key generated by entity i
$\overline{Q}$	denotes the integer obtained from the binary representation of the x-coordinate of an elliptic curve point, Q
H	a cryptographic hash function
$\overline{H}$	an l-bit hash function, $l = (\lfloor \log_2 n \rfloor + 1)/2$
$\|\|$	concatenation symbol
$\oplus$	XOR function
$x \xleftarrow{R} X$	sampling of an element uniformly at random from X

3 ONE - PASS PROTOCOLS

In a one-pass AK protocol it is possible for entities $\hat{A}$ and $\hat{B}$ to agree upon a session key after a single message having been sent from $\hat{A}$ to $\hat{B}$, if $\hat{A}$ has an authenticated copy of $\hat{B}$'s static public key. A two-pass protocol can be converted to one-pass simply by replacing $\hat{B}$'s ephemeral public key with his static public key (Blake-Wilson et al., 1997). In this Section we use precisely this technique create one-pass versions of the following protocols (described in Tables (2 - 7) respectively):

- The Unified Model, proposed by Ankney, Johnson and Matyas (Ankney et al., 1995); it is an AK protocol in the draft standards ANSI X9.42 (ANSI-X9.42, 1998), ANSI X9.63 (ANSI-X9.63, 1998), and IEEE P1363 (IEEE-1363, 1998).
- The Key Exchange Algorithm (KEA) designed by the National Security Agency and declassified in 1998 (NIST, 1998). KEA is the key agreement protocol in the FORTEZZA suite of cryptographic algorithms designed by NSA in 1994 and it is similar to the Goss (Goss, 1990) and MTI/A0 (Matsumoto et al., 1986) protocols.
- The KEA+ protocol proposed by (Lauter and Mityagin, 2001); a modified version of the KEA protocol, which satisfies stronger security requirements than simple KEA for authenticated key-exchange.
- The MQV protocol (Law et al., 1998) that is in the draft standards ANSI X9.42 (ANSI-X9.42, 1998), ANSI X9.63 (ANSI-X9.63, 1998), and IEEE P1363 (IEEE-1363, 1998). MQV was proposed by NSA as the standard key exchange protocol for the US government.
- The HMQV protocol by (Krawczyk, 2005; Menezes, 2005) that was proposed as an alternative of MQV. In particular, there are two one-pass variants, namely HMQV(1) and HMQV(2). The two are quite similar; HMQV(2) was proposed for compatibility reasons (with the others x-round variants of HMQV).

For each protocol, we assume that two entities, say Bob and Alice, own a static key pair, the public part of which is presumed to be known and verified by the other party. Alice generates an ephemeral key pair (r, R) and sends the ephemeral public key, R, to Bob, along with her identity $\hat{A}$. Afterward, they compute a session key which can be shown to be the same for the both of them.

Table 2: One-pass UM.

Alice (a, A)		Bob (b, B)
$r \xleftarrow{R} \mathbb{Z}_n^*$, $R = rP$	$\xrightarrow{R, \hat{A}}$	
$sk_A = aB \|\| rB$		$sk_B = bA \|\| bR$

Table 3: One-pass KEA.

Alice (a, A)		Bob (b, B)
$r \xleftarrow{R} \mathbb{Z}_n^*$, $R = rP$	$\xrightarrow{R, \hat{A}}$	
$sk_A = aB \oplus rB$		$sk_B = bA \oplus bR$

Table 4: One-pass KEA+.

Alice (a, A)		Bob (b, B)
$r \xleftarrow{R} \mathbb{Z}_n^*$, $R = rP$	$\xrightarrow{R, \hat{A}}$	
$sk_A = H(aB, rB, \hat{A}, \hat{B})$		$sk_B = H(bA, bR, \hat{A}, \hat{B})$

4 KEY-COMPROMISE IMPERSONATION ATTACKS

Obviously, if a private key is compromised then the attacker can impersonate the "corrupted" party to other entities, because entities are identified precisely by their private key. This kind of impersonation attack cannot be prevented in any of the existing public key cryptographic schemes. Instead, by "resistance to key-compromise impersonation (K-CI) attacks", we will understand the property of a protocol whereby if one party's long-term private key is somehow disclosed to an adversary, then that adversary will not be

Table 5: One-pass MQV.

Alice (a,A)		Bob (b,B)
$r \xleftarrow{R} \mathbb{Z}_n^*$, $R = rP$	$\xrightarrow{R,\hat{A}}$	
$sk_A = (r+\bar{R}a)(1+\bar{B})B$		$sk_B = (b+\bar{B}b)(R+\bar{R}A)$

Table 6: One-pass HMQV(1).

Alice (a,A)		Bob (b,B)
$r \xleftarrow{R} \mathbb{Z}_n^*$, $R = rP$	$\xrightarrow{R,\hat{A}}$	
$sk_A = (r+ad)B$		$sk_B = (bR+bdA)$
where $d = \bar{H}(R,(\hat{A},\hat{B}))$		

able to impersonate *other entities* to that party (Blake-Wilson et al., 1997). A number of security models for K-CI resilience of AKE protocols have been developed in the literature (Zhu et al., 2005; Krawczyk, 2005; LaMacchia et al.,). The work in (Krawczyk, 2005), mentions that protocols which use long-term static Diffie-Hellman keys g^a, g^b to derive a session key g^{ab} (as all of the one-pass protocols examined here do) are insecure against K-CI attacks, but does not elaborate further. Before describing any attacks, we briefly mention some of the applications for which the use of one-pass protocols has been proposed (Oh et al., 2003), and the consequences of a K-CI attack in each setting.

4.1 Consequences of K-CI Vulnerability

The major danger with K-CI is that an adversary can possibly gain much more knowledge than by simply having access to past or future conversations of an entity. Obviously, with knowledge of a party's private key, an attacker can eavesdrop and decrypt past or future conversations of that party[2]. Besides eavesdropping, however, a KC-I attacker would also be able to actively elicit additional information that may never have been communicated otherwise, by pretending to be a trusted entity to the victim (e.g., the attacker steals one's private key and then is able to pretend to be their lawyer or business associate).

E-mail In an e-mail system one may wish to send encrypted messages by only using their own public information, such as name or e-mail address. Because one party may be temporarily off-line, e-mail communication resembles a one-way channel, and thus an one-pass AK protocol might be suitable in order to send a message without additional communication overload (Oh et al., 2003). All modern one-pass schemes provide assurance that no user other than the receiver will be able to compute the value of the shared secret key, as long as users remain uncorrupted. However, the vast number of e-mail users combined with the extensive presence of malicious software, makes it likely that private keys stored on personal computers (e.g., in conventional memory) can be compromised. Examples of serious K-CI consequences include the impersonation of a government entity or victims's lawyer to obtain information, and the impersonation of a stockbroker's clients and vice-versa.

[2]This attack can be prevented by modern x-round protocols, in which both parties exchange an ephemeral public key.

Table 7: One-pass HMQV(2).

Alice (a,A)		Bob (b,B)
$r \xleftarrow{R} \mathbb{Z}_n^*$, $R = rP$	$\xrightarrow{R,\hat{A}}$	
$sk_A = (1+e)(r+da)B$		$sk_B = (R+dA)(b+be)$
where $d = \bar{H}(R,\hat{B})$ and $e = \bar{H}(B,\hat{A})$		

E- Commerce For transactions held exclusively in cyberspace, one needs a key agreement protocol that offers authentication of the sender's identity. Furthermore, as the session key must be changed in every session, a protocol must provide both implicit key authentication and key freshness. One-pass AK protocols meet both of these requirements, and have been proposed as a possible mechanism for secure e-shopping. The consequences of a K-CI attack on an on-line transaction might include an adversary, say Eve, impersonating an on-line shop to a client whose private key she has obtained, and asking for personal or credit information.

Mobile Transactions In wireless communications, such as wireless e-commerce, the authentication of a user is a very important issue, since its physical location changes frequently. Moreover, the computational power of a mobile device is likely to be limited. In light of these considerations, one-pass AK protocols have been proposed as a possible solution in wireless environments, because of their low communication overhead. As with K-CI attacks on e-commerce applications, an adversary can cause the disclosure of confidential data from the victims. Moreover, in cases where the attacker impersonates the wireless connection server, victims may be connected on an unauthorized network, and thus their (mobile) computer may be corrupted further.

4.2 K-CI Attacks

We will distinguish between two types of K-CI attacks, defined below.

4.2.1 Type-1

All existing one-pass AK establishment protocols are open to the general K-ci attack, in which an intruder, Eve, masquerades as a different entity and tries to establish a valid session key with the compromised party, Bob. There is no need for eavesdropping in this case: Eve, knowing Bob's private key, can initiate a new session with him by creating and sending an ephemeral public key, R, pretending to be another honest entity, Alice. In that case, Eve can compute the same session key as Bob, who is convinced that the key is shared with Alice. The attack is illustrated in Table 8. Its success is based on the fact that none of the one-pass approaches mentioned here includes a sender verification mechanism. For instance, an exponential challenge-response (XCR) signature (from a player A to a player B), used in the HMQV protocol (Krawczyk, 2005), can also be constructed by anyone who has knowledge of the recipient's private key. This means that if an attacker has knowledge of B's private key, he is able to create a signature of this type and thus impersonate A to B.

Table 8: Type-1 K-CI attack on HMQV(1).

Eve knows b, B, A		Bob (b, B)
$r \xleftarrow{R} \mathbb{Z}_n^*$, $R = rP$	$\xrightarrow{R,\hat{A}}$	
$sk_E = (bR + bdA)$		$sk_B = (bR + bdA)$
where $d = \bar{H}(R, (\hat{A}, \hat{B}))$		

Table 9: Solution to Type-1 K-CI attack on HMQV(1).

Alice (a, A)		Bob (b, B)
$r \xleftarrow{R} \mathbb{Z}_n^*$, $R = rP$	$\xrightarrow{R,\hat{A},T,Sig_{\hat{A}}(R,T,\hat{B})}$	
		verify $Sig_{\hat{A}}(R, T, \hat{B})$
		if OK continue
$sk_A = (r + ad)B$		$sk_B = (bR + bdA)$
where $d = \bar{H}(R, (\hat{A}, \hat{B}))$		

A possible solution to the Type-1 K-CI attack would be to have the sender transmit their digital signature on her ephemeral public key (see Table 9). Then, the receiver would be able to verify the signature before accepting the key (and the sender's identity). We stress the importance of including the recipient's identity, $\hat{B}$, in the signed message to avoid the possibility of an attacker impersonating A by reusing A's signature from a protocol run between A and a different entity. The procedure described above does not protect against replay attacks. One way to reduce, but do not eliminate, the replay vulnerability, is to have parties append time-stamps to their messages[3]. More specifically, B can examine the time-stamp T sent by the protocol initiator, A, and terminate the protocol if "too much" time has elapsed since T. Of course, this requires synchronization of A's and B's clocks, to within some reasonable tolerance. Depending on the statistics of the transmission delay imposed by the communication channel, an entity can set a time threshold that leaves a potential attacker little time to mount a replay attack. If A's and B's clocks are perfectly synchronized and the transmission delay is known with certainty, then the time left for an attack could be made arbitrarily small. The question of what is an acceptable time threshold will generally be application-dependent, and will not be discussed further here. Finally, one could also claim that signing every message involving the shared key could be a possible solution to Type-1 K-CI attacks, however, the additional communication/computational cost would be very high.

Remark: We have not included a formal proof of security against Type-1 K-CI attacks for the fix proposed in this Section. Such proof could be constructed based on the enhanced Canneti-Krawczyk model in (Zhu et al., 2005), where in addition to the typical queries an adversary can make, one introduces a new query called *key compromise*. When an adversary issues this query for a specified party, B, the adversary learns B's long-term secret, b, but no other internal information. The key compromise query is different from the weaker type of party corruption query described in (Bellare et al., 2000; Katz et al., 2002) under their "weak-corruption" model, because a party may be uncorrupted while compromised. Furthermore, because in our case there is a single data flow, one can easily show that a successful Type-1 K-CI attack against the protocol in Tab. 4.2.1, for example, implies that the adversary has defeated the digital signature scheme under the assumptions made on the time-stamps T.

4.2.2 Type-2

There is a special K-ci attack that apparently succeeds with *all* one-flow protocols. It is illustrated in Table 10. An intruder, Eve, that learns Bob's secret key and then eavesdrops on a single message from Alice (the initiator of the protocol) to Bob, would then be

[3]We note that the proposed technique for improving K-CI security in HMQV can be made more efficient by computing d as $\bar{H}(R, (\hat{A}, \hat{B}), T)$ and signing only the d value.)

Table 10: Type-2 K-CI attack on HMQV(1).

Alice (a,A)	Eve knows b,B,A	Bob (b,B)
$r \xleftarrow{R} \mathbb{Z}_n^*$, $R = rP$		
$\xrightarrow{R,\hat{A},T,Sig_{\hat{A}}(R,T,\hat{B})}$	$\dashrightarrow$	$\xrightarrow{R,\hat{A},T,Sig_{\hat{A}}(R,T,\hat{B})}$
		verify $Sig_{\hat{A}}(R,T,\hat{B})$
	intercept Alice	
	$sk_E = (bR + bdA)$	$sk_B = (bR + bdA)$
	where $d = \bar{H}(R,(\hat{A},\hat{B}))$	

able to compute the current session key and thus impersonate Alice (but *no one else*) to Bob, and only for the current session. To achieve this, after Eve intercepts Alice's ephemeral public key, R, she computes the session key in the same way as Bob, and then must "cut out" Alice from the current conversation. There is no apparent solution for this attack, even if a scheme is to be equipped with digital signatures or time-stamps, or both. However, the Type-2 attack is rather limited compared with the general K-CI attack in which the intruder can impersonate *any* entity and at *any* time.

5 CONCLUSIONS

In this paper we have examined the resistance of the most efficient one-pass asymmetric AK establishment protocols to K-CI attacks. The use of one-pass protocols is unavoidable in settings where the communication channel is one-way (e.g., e-mail, store-and-forward applications) or in cases where computational and communication cost is to be minimized (e.g., low-power mobile applications). We distinguished between two types of K-CI threats. Unfortunately, due to their similarities, none of the protocols examined here are resistant to either K-CI attack. However, their security against Type-1 K-CI attacks can be somewhat improved with the help of standard digital signatures and time-stamps, at a significant additional communication and computational cost.

Although forward secrecy (another harmful threat related to party corruption) is usually considered more important than K-CI, our discussion suggests that a K-CI attack can be more dangerous: in widely-used applications, such as e-mail, mobile and e-business transactions, the security practices of the average user are likely to be lax (making key-compromise a real possibility) while at the same time a K-CI adversary can ask for and obtain information that would have not been transmitted otherwise. For this reason, the use of one-pass protocols should be avoided when possible.

REFERENCES

Ankney, R., Johnson, D., and Matyas, M. (1995). The unified model. In *Contribution to X9F1*.

ANSI-X9.42 (1998). Agreement of symmetric algorithm keys using Diffie-Hellman. In *Working Draft*.

ANSI-X9.63 (1998). Elliptic curve key agreement and key transport protocols. In *Working Draft*.

Bellare, M., Pointcheval, D., and Rogaway, P. (2000). Authenticated key exchange secure against dictionary attacks. In *Proceedings EUROCRYPT 2000, LNCS 1807, pp. 139-155*. Springer-Verlag.

Bird, R., Gopal, I., Herzberg, A., Janson, P., Kutten, S., Molva, R., and Yung, M. (1991). Systematic design of two-party authentication protocols. In *Proceedings of Advances in Cryptography - Crypto '91, LNCS 576, pp. 44-61*. Springer-Verlag.

Blake-Wilson, S., Johnson, D., and Menezes, A. (1997). Key agreement protocols and their security analysis. In *Proceedings of 6th IMA International Conference on Cryptography and Coding, LNCS 1355, pp. 30-45*. Springer-Verlag.

Blake-Wilson, S. and Menezes, A. (1998). Authenticated Diffie-Hellman key agreement protocols. In *Proceedings of the 5th annual international workshop - SAC '98, pp. 339-361*. Springer-Verlag.

Boyd, C., Mao, W., and Paterson, K.-G. (2004). Key agreement using statically keyed authenticators. In *Proceedings of Applied Cryptography and Network Security - ACNS '04, LNCS 3089, pp. 248-262*. Springer-Verlag.

Diffie, W. and Hellman, M. (1976). New directions in cryptography. In *IEEE Transactions on Information Theory 22(6), pp. 644-654*.

Goss, K.-C. (1990). Cryptographic method and apparatus for public key exchange with authentication. In *U.S. Patent 4956865*.

IEEE-1363 (1998). Standard specifications for public key cryptography. In *Working Draft*.

Jeong, I., Katz, J., and Lee, D. (2004). One-round protocols for two-party authenticated key exchange. In *Applied*

Cryptography and Network Security - ACNS 2004, pp. 220–232., Vol. 3089/2004 of LNCS. Springer-Verlag.

Kaliski, B. (2001). An unknown key share attack on the mqv key agreement protocol. In *ACM Transactions on Information and System Security, pp. 3649*. Springer-Verlag.

Katz, J., Ostrovsky, R., and Yung, M. (2002). Forward secrecy in password-only key exchange protocols. In *Proceedings SCN 2002, LNCS 2576, pp. 29-44*. Springer-Verlag.

Krawczyk, H. (2005). Hmqv: A high-performance secure diffie- hellman protocol. In *Proceedings of Advances in Cryptology - Crypto '05, LNCS 3621, pp. 546-566*. Springer-Verlag.

Kwon, T. (2001). Authentication and key agreement via memorable password. In *NDSS 2001 Symposium Conference Proceedings*.

LaMacchia, B., Lauter, K., and Mityagin, A. Stronger security of authenticated key exchange. http://citeseer.ist.psu.edu/lamacchia06stronger.html.

Lauter, K. and Mityagin, A. (2001). Authentication and key agreement via memorable password. In *NDSS 2001 Symposium Conference Proceedings*.

Law, L., Menezes, A., Qu, M., Solinas, J., and Vanstone, S. (1998). An efficient protocol for authenticated key agreement. In *Technical report CORR 98-05, University of Waterloo*.

Lu, R., Cao, Z., Su, R., and Shao, J. (2005). Pairing-based two-party authenticated key agreement protocol.

Matsumoto, T., Takashima, Y., and Imai, H. (1986). On seeking smart public-key distribution systems. In *Transactions of the IECE of Japan, E69, pp. 99-106,*.

Menezes, A. (2005). Another look at HMQV. In *Cryptology ePrint Archive, Report 2005/205*.

NIST (1998). Skipjack and kea algorithm specification.

Oh, S., Kwak, J., and Lee, S.and Won, D. (2003). Security analysis and applications of standard key agreement protocols. In *ICCSA (2), pp.191-200*. Springer-Verlag.

Strangio, M.-A. (2006). On the resilience of key agreement protocols to key compromise impersonation. In *European PKI Workshop on Public Key Infrastructure , LNCS 4043, pp. 233-247*. Springer-Verlag.

Zhu, R. W., Tian, X., and Wong, D. S. (2005). Enhancing ck-model for key compromise impersonation resilience and identity-based key exchange. Cryptology ePrint Archive, Report 2005/455. http://eprint.iacr.org/.