SCIS2000 The 2000 Symposium on Cryptography and Information Security
Okinawa, Japan, January 26-28, 2000

On the Significance of Unknown Key-Share Attacks : How To Cope With Them?

Joonsang Baek * mohi@icu.ac.kr
Kwangjo Kim * kkj@icu.ac.kr
Tsutomu Matsumoto † tsutomu@mlab.jks.ynu.ac.jp

Abstract— At PKC'99, Blake-Wilson and Menezes [3] proposed a new unknown key-share attack by exploiting a property, called duplicate-signature key selection, that different public keys corresponding to the given signature are efficiently generated. We apply this attack to the key agreement protocol provably secure against active attacks, which were proposed by Hirose and Yoshida at PKC'98. We also extend the idea of duplicate-signature key selection to the variations of ElGamal signature schemes and Schnorr signature scheme.

1 Introduction

An unknown key-share (UKS) attack on a key agreement protocol is characterized by an attack whereby an entity A finishes believing she shares a key with B, and although this is in fact the case, B mistakenly believes that the key is instead shared with an entity $E \neq A$ [3, 4, 6].

A scenario where an UKS attack can have damaging consequences is the following [6]. Let B denote a bank branch and A denote an account holder. Certificates are issued by the bank headquarters and the account information of the holder is contained in each certificate. Suppose that the protocol for electronic deposit of funds is to exchange a key with a bank branch via an authenticated key agreement with key confirmation (AKC) protocol. At the end of the protocol run, encrypted funds are deposited to the account number in the certificate. Suppose that no further authentication is done in the encrypted message. If the UKS attack is successfully launched, then the deposit will be made to E's account instead of A's account. Although UKS attacks are well-known, it has discovered that some proposed key agreement protocols are susceptible to this attacks[3].

In this paper, we apply and generalize Blake-Wilson and Menezes's new UKS attacks exploiting a certain undesirable property of signature schemes. The paper is organized as follows. We introduce the new UKS attacks proposed by Blake-Wilson and Menezes at PKC'99 in Section 2. In Section 3, Hirose and Yoshida's key agreement protocol is described in brief and a possible UKS attack on their protocol is described in the following Section. In Section 5 some countermeasures of preventing Hirose and Yoshida's key agreement protocol from the UKS attack are discussed. In Section 6, we discuss duplicate-signature key selection property rather more and apply the idea of duplicate-signature key selection to the variations of ElGamal signature scheme[9] and Schnorr signature scheme[15]. Concluding remarks will follow in Section 7.

2 New UKS Attack

It is well known that STS (Station-To-Station) protocol is insecure against public key substitution UKS attacks. Before discussing new UKS attacks, we describe well-known public key substitution attack on STS protocol against responder.(The attack can be applied to the case of initiator in a similar manner.)

In this attack as depicted in Fig.(1), the adversary E registers A's public key P_A as its own ; that is, $P_E = P_A$. When A sends B message in the flow (1), E intercepts it and replaces the identity I_A with I_E in the flow (1'). The notation $A \hookrightarrow B$ means that though A transmitted a message intended for B, it was intercepted by the adversary and not delivered to B.

E then passes message from B to A unchanged

* School of Engineering, Information and Communications Univ., 58-4 Hwaam-dong, Yusong-gu, Taejon, 305-348, Korea
† Division of Artificial Environment and Systems, Yokohama National Univ., 79-5 Tokiwadai, Hodogaya-ku, Yokohama 240-8501 Japan,

(1)	$A \hookrightarrow B$	I_A, g^{r_A}
(1')	$E \rightarrow B$	I_E, g^{r_A}
(2)	$E \leftarrow B$	$Cert_B, g^{r_B}, S_B(g^{r_B}, g^{r_A})$, $MAC_K(S_B(g^{r_B}, g^{r_A}))$
(2')	$A \leftarrow E$	$Cert_B, g^{r_B}, S_B(g^{r_B}, g^{r_A})$, $MAC_K(S_B(g^{r_B}, g^{r_A}))$
(3)	$A \hookrightarrow B$	$Cert_A, S_A(g^{r_A}, g^{r_B})$, $MAC_K(S_A(g^{r_A}, g^{r_B})$
(3')	$E \rightarrow B$	$Cert_E, S_A(g^{r_A}, g^{r_B})$, $MAC_K(S_A(g^{r_A}, g^{r_B})$

Figure 1: Public Key Substitution UKS Attack on STS Protocol

(in the flow (2) and (2').) Finally E intercepts message from A to B in the flow (3), and replaces $Cert_A$, which contains A's identifying information, and A's public signature key P_A with $Cert_E$ in the flow (3'). Since $P_A = P_E$, the signature $s_A (= S_A(g^{r_A}, g^{r_B}))$ is verified as correct. Of course, g is generator of a certain multiplicative group in which Diffie-Hellman assumption is hold and r_A and r_B are secret exponents of A and B, respectively. Hence B accepts the key K and falsely believes that K is shared with E, while in fact it is shared with A.

However, the public key substitution attack can be easily prevented by requiring that entities prove to CA (Certificate Authority) possession of the private keys corresponding to their public keys during the certification issuing process.

In 1999, Blake-Wilson and Menezes[3] proposed a new on-line UKS attack on STS protocol. This new UKS attack is based on the following assumption. If P_A (A's public key) and A's signature s_A on a message M are known, an adversary is able to select a key pair (P_E, S_E) with respect to which s_A is also E's signature on the message M. This new UKS attack on STS protocol can be similarly described to the public key substitution attack against responder. After A sends message in the flow(3), E intercepts it and selects a key pair (P_E, S_E) so that s_A is also E's signature on message (g^{r_A}, g^{r_B}), where S_E are E's private signing keys for a signature scheme. E then obtains a certificate $Cert_E$ for P_E, and sends message with the certificate to B in the flow(3'). This property of the signature scheme is called *duplicate-signature key selection(DSKS)*[3]. In this paper, we call an UKS attack exploiting DSKS property of the employed signature scheme "DSKS UKS attack".

Before further discussing DSKS property in later sections, we remark that it was shown that conventional signature schemes such as RSA[14], DSA[7], ElGamal, Rabin[13], and ECDSA[8] possess the DSKS property in [3]. Also, we note that E is able to get his public key certified during a run of the STS protocol. This is plausible since delays in the transmissions of message sometimes happens where CA is on-line.

The significance of DSKS property is that it is useless for CA to check the entities possess the private keys corresponding to the public keys when issuing certificates. However, it should be noted that DSKS property does not always mean that the underlying signature scheme is insecure. Usually, the security of signature schemes implies that they are at least secure against existential forgery.

3 Hirose and Yoshida's Key Agreement Protocol

At PKC'98, Hirose and Yoshida proposed a key agreement protocol. We call their protocol KAP-HY98. They proved that KAP-HY98 is secure against active adversary in the random oracle model[11] : It was shown to be secure against passive eavesdropping, impersonation, interference, active eavesdropping, and pretense. Their security analysis is unique in the sense that they did not model the distributed environments as done in [1, 2], but formally defined various attacks and showed their scheme is secure against the defined attacks. To provide key confirmation for their key agreement protocol, they proposed a modification of Schnorr signature scheme, called a *redundant signature scheme.* A brief description of the scheme is following.

Redundant Signature Scheme

Let $S_B \in \mathrm{Z}_q$ be a secret of user B who publishes $v_B = g^{-S_B} \bmod p$. Let $h : \{0,1\}^* \rightarrow \mathrm{Z}_q$ be a collision free hash function. Let $k_B \in \mathrm{Z}_q$ and $u_B = g^{-k_B} \bmod p$. Suppose that B knows k_B. B's signature (u_B, e_B, w_B) for u_A is computed as follows :

1. B selects $r_B \in \mathrm{Z}_q$ at random and computes $x_B = g^{r_B} \bmod p$.
2. B computes $e_B = h(x_B, u_B, u_A)$ and $w_B = r_B + e_B k_B + e_B^2 S_B \bmod q$.

The verification procedure for the signature (u_B, e_B, w_B) for u_A is as follows.

1. Let $z_B = g^{w_B}(u_B v_B^{e_B})^{e_B} \bmod p$.

2. The signature is valid if and only if $e_B = h(z_B, u_B, u_A)$.

Now, we describe how KAP-HY98 works in brief. It is assumed that the public information of each participant, for example, a certificate is contained in his identity, I_A(or I_B).

KAP-HY98

1. A selects at random $k_A \in \mathrm{Z}_q$ and computes $u_A = g^{-k_A} \bmod p$. A sends u_A and I_A, where I_A is identification of A, to B.

2. B selects $k_B \in \mathrm{Z}_q$ and computes $u_B = g^{-k_B} \bmod p$. Then B computes a signature (u_B, e_B, w_B) for u_A using redundant signature scheme. B sends u_B, e_B, w_B, and I_B, where I_B is identity of B, to A.

3. A checks the validity of signature (u_B, e_B, w_B). If it is not valid, A terminates the protocol. Otherwise, A computes a signature (u_A, e_A, w_A) for u_B using the redundant signature scheme and sends e_A, and w_A to A. A also computes $K_A = u_B^{-k_A} \bmod p$. K_A will be a session key.

4. B checks the validity of signature (u_A, e_A, w_A). If it is not valid, A terminates the protocol. Otherwise, computes $K_B = u_A^{-k_B} \bmod p$, which becomes the session key between A and B.

4 UKS attack on KAP-HY98

In this section we show that the KAP-HY98 is vulnerable to both the public key substitution and the DSKS UKS attack. Before describing the attack, note that originally, the KAP-HY98 was shown to be secure against UKS attack. Instead of calling 'UKS attack', Hirose and Yoshida used the term 'pretense'. However we point out that their proof did not consider other case of pretense.

Let us examine the following UKS attack on the KAP-HY98 already described in [11] as shown in Fig.(2). The notations used in this Section are the same ones as in Section 2.

If the above attack were launched, the adversary E would make B believe that the key is shared between B and E. The attack is in fact impossible since the modification of the signed message (e_A, w_A) to (e_E, w_E) in the flow (3'a) implies solving discrete logarithm problem of the signature scheme as proved in [11].

$$
\begin{array}{lll}
\text{(1a)} & A \hookrightarrow B & u_A, I_A \\
\text{(1'a)} & E \rightarrow B & u_A, I_E \\
\text{(2a)} & E \leftarrow B & (u_B, e_B, w_B), I_B \\
\text{(2'a)} & A \leftarrow E & (u_B, e_B, w_B), I_B \\
\text{(3a)} & A \hookrightarrow B & e_A, w_A \\
\text{(3'a)} & E \rightarrow B & e_E, w_E
\end{array}
$$

Figure 2: UKS attack on KAP-HY98

But if the attack is launched in reverse direction, that is, against initiator, the situation becomes rather different. In this attack as shown in Fig.(3), the adversary E registers B's public key P_B as its own; that is, $P_E = P_A$. When B sends A in the flow (2b), E intercepts it and changes the certificate of P_B into that of P_E. E then passes message in the flow (3b) from A to B unchanged. This is a public key substitution UKS attack.

$$
\begin{array}{lll}
\text{(1b)} & A \rightarrow E & u_A, I_A \\
\text{(1'b)} & E \rightarrow B & u_A, I_A \\
\text{(2b)} & A \hookleftarrow B & (u_B, e_B, w_B), I_B \\
\text{(2'b)} & A \leftarrow E & (u_B, e_B, w_B), I_E \\
\text{(3b)} & A \rightarrow E & e_A, w_A \\
\text{(3'b)} & E \rightarrow B & e_A, w_A
\end{array}
$$

Figure 3: New UKS attack on KAP-HY98

Even if the public key substitution attack is not possible to launch, the DSKS UKS attack may be possible: The adversary E passes message in the flow (1) unchanged. After B sends message in the flow (2b), the adversary E intercepts it and selects a public key and a secret key pair (P_E, S_E) for the signature (u_B, e_B, w_B) for u_A so that (u_B, e_B, w_B) is also E's signature on the message in the flow (2b'). E then obtains a certificate for P_E and sends this after putting in I_E (I_E contains P_E and its certificate). Again, E passes message from A to B unchanged.

Next, we show that the redundant signature scheme possesses DSKS property. For the formal treatment of this problem, we give definitions of DSKS as below.

Definition 1 *(Advantage of Adversary) Let E be an adversary and define the advantage of E in selecting duplicate-signature key as following probability.*

$$
\begin{aligned}
Adv_{Sig}^{DSKS}(E) \quad = \quad & \Pr[\mathcal{V}_{(P_E)}(M, \sigma_A) = 1 : \\
& (P_A, S_A) \xleftarrow{R} K; M \xleftarrow{R} \{0,1\}^*;
\end{aligned}
$$

$$\sigma_A \leftarrow Sig_{S_A}(M);$$
$$(P_E, S_E) \leftarrow E(P_A, S_A, \sigma_A)]$$

Here, (P_A, S_A) denotes A's public and private key pair. σ_A is signature of message M, $\mathcal{V}$ and Sig are verification and signature procedures, respectively. (P_E, S_E) is E's new public and private key pair.

Now, DSKS property is defined as follows.

Definition 2 *(DSKS Property) If $Adv_{Sig}^{DSKS}(E)$ is non-negligible, we say that the signature scheme has a DSKS property.*

Then, we can obtain the interesting result on the redundant signature scheme.

Theorem 1 *The redundant signature scheme possesses the DSKS property.*

Proof: Let $P_B(= (p, g, v_B))$ be a public parameter for the redundant signature scheme. Here, g is a generator of the multiplicative group Z_p^* and $v_B = g^{-S_B} \bmod p$, where S_B is kept as secret. Now let $\sigma_B = (u_B, e_B, w_B)$ be a signature for u_A. Suppose that $u_B = g^{-k_B} \bmod p$ for the secret k_B chosen by B, $e_B = h(x_B, u_B, u_A)$ where $x_B = g^{r_B} \bmod p$, h is a collision resistant hash function, and $w_B = r_B + e_B k_B + e_B^2 S_B \bmod q$ as described in Section 3.

The adversary E can select another public parameter $P_E = (p, \bar{g}, \bar{v}_B)$ for the signature σ_B as follows. Given P_A and σ_B, E chooses l at random, where $2 < l < p-1$ and $gcd(l, p-1) = 1$ and computes $\bar{g} = g^l \bmod p$. Now he computes $\bar{v}_B = g^{(1-l)w_B/e_B^2} v_B$. Then $P_E = (p, \bar{g}, \bar{v}_B)$ becomes another public parameter for the signature s_B such that $P_E \neq P_A$, because

$$\begin{aligned}
& (\bar{g})^{w_B}(u_B \bar{v}_B^{e_B})^{e_B} \\
= \ & (g^l)^{w_B} g^{-k_B e_B} (g^{(1-l)w_B/e_B^2} v_B)^{e_B^2} \\
= \ & g^{l w_B} g^{-k_B e_B} g^{(1-l)w_B} g^{-S_B e_B^2} \\
= \ & g^{l w_B - k_B e_B + w_B - l w_B - S_B e_B^2} \\
= \ & g^{w_B - k_B e_B - S_B e_B^2} \\
= \ & g^{r_B + e_B k_B + e_B^2 S_B - k_B e_B - S_B e_B^2} \\
= \ & g^{r_B} \\
= \ & x_B
\end{aligned}$$

That is, the signature scheme is still verified as correct. Hence we can conclude that E can select duplicate-signature key with non-negligible probability. ¶

By **Theorem 1**, E is able to mount the UKS attack on the KAP-HY98 : Though A believes that he is sharing a key with entity E, he actually shares the key with entity B.

5 Thwarting the Attack

We have shown that the KAP-HY98 is insecure under UKS attacks. There are some countermeasures to prevent the attack. We note that these countermeasures are not only applied to KAP-HY98 but also generally applied to other key agreement protocols.

(A) If certificates of both entities are exchanged prior to the key agreement protocol, then UKS attacks fail. However, it is not desirable in the situation that requires low bandwidths.

(B) If CA issues certificates with checking that the entities possess the private keys corresponding to their public keys, it is impossible to mount UKS attack. But this is not so desirable as well since no one guarantees that every CA will conduct the checking procedure. Also, if the signature scheme used for key confirmation has DSKS property it is of no use for CA to check that the entities possess the private keys corresponding to the public keys.

(C) Including the identities of the sender and receiver in the messages being signed prevents UKS attack. This method is desirable since there will be no additional overhead on the run of key agreement protocols.

As mentioned above, the method **(C)** is considered to be a good countermeasure on UKS attacks. For example, if the identities of A and B are included in the messages being signed in the redundant signature scheme, the UKS attack will be prevented: The signature $e_A = h(x_A, u_A, u_B)$ is changed to $e'_A = h(x_A, u_A, u_B, I_A)$. Similarly, $e_B = h(x_B, u_B, u_A)$ is changed to $e'_B = h(x_B, u_B, u_A, I_B)$. The significance of this methods are discussed in [3, 10, 12, 16].

6 More On DSKS Property

It may be interesting to find a way to remove DSKS property from the existing signature schemes.

In [3], it was not generally shown whether every signature scheme, at least, every ElGamal–type signature scheme has DSKS property. We show that some variations of ElGamal signature scheme possess the DSKS property. Here, we restrict some variations of original ElGamal signature scheme that satisfy signing equation $u = av + kw \bmod p-1$ where the public key is $y = g^a \bmod p$. The possible values of u, v and w are $h(m), r$ and s. Here, h is a hash function, m is a message, $r(=g^k)$ and s are signatures of $h(m)$. Table 1 lists the 6 possibilities of signing equation and their corresponding verification algorithms.

(u,v,w)	Signing Equation	Verification
$(h(m),r,s)$	$h(m) = ar + ks$	$g^{h(m)} = (g^a)^r r^s$
$(h(m),s,r)$	$h(m) = as + kr$	$g^{h(m)} = (g^a)^s r^r$
$(s,r,h(m))$	$s = ar + kh(m)$	$g^s = (g^a)^r r^{h(m)}$
$(s,h(m),r)$	$s = ah(m) + kr$	$g^s = (g^a)^{h(m)} r^r$
$(r,s,h(m))$	$r = as + kh(m)$	$g^r = (g^a)^s r^{h(m)}$
$(r,h(m),s)$	$r = ah(m) + ks$	$g^r = (g^a)^{h(m)} r^s$

Table 1: Variations of ElGamal Signature Scheme

Theorem 2 *The variations of ElGamal signature scheme possess DSKS property.*

Proof: Note first that $r = g^k$ where g is a generator of multiplicative group z^* and k is a secret. Generally, a verification algorithm is written as

$$g^u = y^v r^w = y^v (g^k)^w \text{where } y = g^a.$$

An adversary E selects a duplicate-signature key as follows. Let $\bar{g}$ and $\bar{y}$ denote public key-parameters chosen by E, respectively. If $P_E = (\bar{g}, \bar{y})$ is to be the E's own public key, he must select g^l for $\bar{g}$ and $\bar{g}^{a'} = g^{la'}$ for $\bar{y}$ since $\bar{g}$ and $\bar{y}$ must satisfy the above verification algorithm and the signature r in the verification algorithm must not be changed. Here, the random l and a' satisfy that $2 < l, a' < p-1$ and $gcd(l, p-1) = gcd(a', p-1) = 1$. That is, $\bar{g}$ and $\bar{y}$ satisfy that

$$g^{lu} = \bar{g}^u = \bar{y}^v (g^k)^w = g^{la'v} (g^k)^w = g^{la'v} r^w.$$

Hence if we take $\bar{g}$ as g^l, we have

$$\bar{y} = g^{la'} \quad = \quad g^{(lu-kw)/v} = (\bar{g})^{u/v} r^{-w/v}.$$

Note that since u, v, and w are publicly known and hence E can generate $\bar{g}$ and $\bar{y}$. (Alternatively we can take $\bar{g}$ as $r^{w/(u-a'v)}$ and $\bar{y}$ as $\bar{g}^{a'}$.) Hence E can select public key for given signature scheme with non-negligible probability. ¶

The proof of **Theorem 2** leads us to get a generalized result. If an adversary wants to generate a valid new public key of given signature, he should choose a new generator $\bar{g}$ replacing the previous generator g and a new public key $\bar{y}$ replacing y. But $\bar{g}$ and $\bar{y}$ must pass the verification procedure of the given signature. Hence he can select the new generator and the public key by solving equation of the verification algorithm parameterized by $\bar{g}$ and $\bar{y}$. In Theorem 2, E selects $\bar{g}$ and $\bar{y}$ that satisfy $\bar{g}^u = \bar{y}^v (r)^w$, which is an equation for verification procedure. If he lets $\bar{g}$ be g^l for some l and $\bar{y}$ be $g^{la'}$ respectively, he find the value, $g^{la'}$ which satisfy $g^{lu} = g^{la'v} r^w$. Since u, v, and w are known and l is chosen by E, he can figure out the value $g^{la'}$.

Now, we consider Schnorr signature scheme. The public key of Schnorr's scheme is $y = g^{-a}$. If one signs message m, he selects $k \in \mathrm{Z_q}$ at random and computes $h(x, m)$ where h is cryptographic hash function. Then he computes $e = h(x, m) \bmod q$ and $s = ae + k \bmod q$. (e, s) is a signature for m. To verify the signature (e, s), compute $x' = g^s y^e$ and check that $e = h(x', m)$.

Now, suppose that adversary E chooses his own public key. Like previous examples, he uses $x' = g^s y^e$ to find new $\bar{g}$ and $\bar{y}$: He solves the equation $\bar{g}^s \bar{y}^e = g^s y^e$. If he puts $\bar{g} = g^l$ and $\bar{y} = g^{la'}$, he gets new public key $\bar{y} = g^{s(1-l)/e} y^{-1}$. This result gives us the following theorem.

Theorem 3 *Schnorr signature scheme possesses the DSKS property.*

From above results, we can see that many discrete logarithm based- signature schemes have DSKS property. At least, it is a fact that adversary can generate his new public key of the signature schemes in which signing equation is linear combination of secret values and verification is calculated by exponentiationing public values by signature values with the same method described in the proofs of **Theorems 2** and **3**. However, constructing a signature scheme which does not have DSKS property may be an interesting research topic. Since the adversary could generate a new public key and its corresponding secret key by solving a verification algorithm in which known signature parameters are inputted as an equation , the attack was always successful. But if the verification algorithm is not solely proceeded with known parameters, the attack can be avoided. Hence such signature scheme as the undeniable signature[5] that verification is proceeded by co-operation of signer can prohibit DSKS property.

we remain formal justification of this idea as future research.

7 Concluding Remarks

We have shown that KAP-HY98 is vulnerable to UKS attack. Also, we have generalized the results in [3] and applied to various ElGamal-type signature and Schnorr signature schemes. Though it is not fully generalized in this paper, it is conjectured that all discrete-log based signature schemes possess the DSKS property. Also, whether it is possible to construct DSKS-free signature scheme is another open problem.

It should be noted that the prevention of UKS attacks is generic property of the formal models of distributed environments proposed in [2, 3]. Although which model effectively describes the distributed environments for secure key agreement protocol is still controversial, we consider that constructing formal models of key agreement protocols is more sound approach than case-by-case analysis of possible attacks.

References

[1] M. Bellare, R. Canetti, and H. Krawczyk, "A Modular Approach to the Design and Analysis of Authentication and Key Exchange Protocols", *Proceedings of Annual Symposium on the Theory of Computing*, 1998, A full version is available at `http://www-cse.ucsd.edu/users/mihir`

[2] M. Bellare and P. Rogaway, "Entity Authentication and Key Distribution", *Advances in Cryptology – Proceedings of Crypto'93, LNCS 773*, pp.232–249, Springer-Verlag, 1993.

[3] S. Blake-Wilson and A. Menezes, "Unknown Key-Share Attacks on the Station-to-Station (STS) Protocol", *Public Key Cryptography – Proceedings of PKC'99, LNCS 1560*, pp.154–170, Springer-Verlag, 1999.

[4] S. Blake-Wilson and A.Menezes, "Authenticated Diffie-Hellman Key Agreement Protocols", *Selected Areas in Cryptography – Proceedings of SAC'98, LNCS 1556*, pp.339–361, Springer-Verlag, 1999.

[5] D. Chaum and H. V. Antwerpen, "Undeniable Signatures", *Advances of Cryptology – Proceedings of Crypto'89, LNCS 435*, pp.212–216, Springer-Verlag, 1990.

[6] W. Diffie, P. van Oorschot and M. Wiener, "Authentication and Authenticated Key Exchanges", *Designs, Codes and Cryptography*, pp.107–125, 1992.

[7] ANSI X9.30(Part 1), "Public Key Cryptography Using Irrevisible Algorithms for the Financial Service Industry - Part 1 : The Digital Signature Algorithm (DSA)", 1995.

[8] ANSI X9.62, "The Elliptic Curve Digital Signature Algorithm(ECDSA)",working draft,March 1995.

[9] T. ElGamal, "A Public Key Cryptosystem and a Signature Scheme based on Discrete Logarithms", *IEEE Trans. on Information Theory*, IT-31(4), pp.489–472, July, 1985.

[10] G. Horn and P. Preneel, "Authentication and Payment in Future Mobile Systems", *Computer Security - Proceedings of ESORICS'98, LNCS 1485*, pp.277–293, Springer-Verlag, 1998.

[11] S. Hirose and S. Yoshida, "An Authenticated Diffie-Hellman Key Agreement Protocol Secure Against Active Attacks", *Public Key Cryptography – Proceedings of PKC'98, LNCS 1431*, pp.135–148, Springer-Verlag, 1998.

[12] C. Mitchell and A. Thomas, "Standardising Authentication Protocols Based on Public Key Techniques", *Journal of Computer Security*, pp.23–46,1993.

[13] M. O. Rabin,"Digitalized Signatures and Public-Key Functions as Intractable as Factorization ", *MIT/LCS/TR-212*, MIT Laboratary for Computer Science, 1979.

[14] R. L. Rivest, A. Shamir, L. M. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems ", *Communications of the ACM, 21* , pp.120-126, 1978.

[15] C. P. Schnorrr, "Efficient Identification and Signatures for Smart Cards ", *Advances in Cryptology - Proceedings of Crypto'89, LNCS 435*, pp.239–252,1990.

[16] V. Shoup, "On Formal Models for Secure Key Exchange", *IBM Research Report RZ 3120*, 1999, A full version is available at `http://www.shoup.net/papers`