LETTER

Remarks on the Unknown Key-Share Attacks*

Joonsang BAEK†, *Nonmember and* **Kwangjo KIM**†, *Member*

SUMMARY There have been two types of *UKS(unknown key-share)* attacks in the literature: One is the public key substitution UKS attack and the other is the new type of UKS attack presented at PKC'99 by Blake-Wilson and Menezes. This new type of UKS attack exploits a certain property of signature schemes that different public keys corresponding to the given signature are efficiently generated.

In this letter, some flaws in the previous works on UKS attacks are pointed out. Although Blake-Wilson and Menezes proposed a countermeasure for preventing the new type of UKS attack and applied it to revise STS-MAC(Station-to-Station Message Authentication Code) protocol [4], it is shown that the resulting protocol is still vulnerable to the new UKS attack. Also, Hirose and Yoshida's key agreement protocol [8] presented at PKC'98 is shown to be insecure against public key substitution UKS attacks. Finally, countermeasures for such UKS attacks are discussed.

key words: Unknown key-share attack, Duplicate-Signature Key Selection, Public Key Substitution, Key Agreement Protocol.

1. Introduction

An UKS attack on key agreement protocols is an attack whereby an entity A finishes an execution of a key agreement protocol believing that a common key is shared with an entity B (this is in fact the case) but B falsely believes that the key is shared with another entity E ($\neq A$).

The attack scenario first described in [6] is as follows: Suppose that B is a bank branch and A is an account holder. Certificates are issued by the bank headquarters and identifying information of the holder is contained in each certificate. Also, suppose that the protocol for electronic deposit of funds involves exchanging a key with a bank branch via a key agreement protocol. At the end of the protocol execution, encrypted funds are deposited into the account number in the certificate. Assume that no further authentication is done in the encrypted deposit message. If the UKS attack described above is successful, then the deposit will be made to an adversary E's account instead of A's account.

In this letter, we point out some flaws of previous works on UKS attacks and suggest countermeasures against them. The organization of this letter is as follows: Two types of the proposed UKS attacks are reviewed in Sect. 2. In Sect. 3, we show that the revised STS-MAC protocol proposed by Blake-Wilson and Menezes to prevent such UKS attacks is still vulnerable to a certain kind of UKS attacks. Also, Hirose and Yoshida's key agreement protocol [8] is shown to be insecure against public key substitution UKS attacks. In Sect. 4, countermeasures for preventing UKS attacks are discussed and concluding remarks will follow in Sect. 5.

†The authors are with Information and Communications Univ., Taejon 305-348, Korea

*A part of this letter was presented at SCIS2000-C29.

2. Two Types of UKS Attacks

It is known the original STS-MAC protocol is insecure against the public key substitution UKS attack. We describe the well-known public key substitution attack on STS-MAC protocol against the responder. The attack can be applied to the case of the initiator in a similar manner. Throughout this letter, the notation $A \hookrightarrow B$ means that although A transmitted a message intended for B, it was intercepted by an adversary and not delivered to B. α is a generator of a given multiplicative group in which Diffie-Hellman assumption holds.

In the public key substitution UKS attack depicted in Fig. 1, an adversary E registers A's public key P_A as its own, i.e., $P_E = P_A$. When A sends B the message in flow (1a), E intercepts it and replaces the identity I_A with her identity I_E in flow 1'a. Note that, in Fig. 1, r_A and r_B are secret values of the entities A and B, respectively.

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

Fig. 1 Public Key Substitution UKS Attack on STS-MAC Protocol.

E then passes the message from B to A unchanged in flows (2a) and (2'a). Finally, E intercepts the message from A to B in flow (3a), and replaces $Cert_A$, which contains A's identity and the public key P_A for signature verification, with $Cert_E$ in flow (3'a). Since $P_A = P_E$, the signature $S_A(\alpha^{r_A}, \alpha^{r_B})$, where S_A is an

A's signing private key, is verified as being correct.

However, we note that this public key substitution UKS attack can be prevented by requiring CA(Certificate Authority) to check that each entity possesses a private key corresponding to its public key during the certificate issuing process. But, this does not completely prevent UKS attacks as will be described subsequently.

In 1999, Blake-Wilson and Menezes [4] proposed a new type of the on-line UKS attack on STS-MAC protocol. This new UKS attack, which we call "DSKS-UKS" attack, is based on the following DSKS(Duplicate-Signature Key Selection) property of signature schemes: If P_A (A's public key) and A's signature σ_A on a message M are known, an adversary is able to select a public-and-private key pair (P_E, S_E) with respect to which σ_A is also E's signature on the message M.

A DSKS-UKS attack on STS-MAC protocol is similar in description to the public key substitution attack against the responder. After A sends the message in flow (3a), E intercepts it and selects a public-and-private key pair (P_E, S_E) so that $S_A(\alpha^{r_A}, \alpha^{r_B})$ is also E's signature on the message $(\alpha^{r_A}, \alpha^{r_B})$. E then obtains a certificate $Cert_E$ for P_E, and sends the message with the certificate to B in flow (3'a).

We note that conventional signature schemes such as RSA [12], DSA [1], ElGamal [7], Rabin [11], and ECDSA [2] were shown to possess the DSKS property in [4]. Also, note that E is assumed to be capable of obtaining her certified public key during an execution of STS-MAC protocol, i.e., this attack is on-line. This assumption is plausible since delays can occur during transmission of messages even though CA is on-line. The significance of the DSKS property relies on the fact that CA's checking procedures for ensuring whether each entity possesses a private key corresponding to its public key can be useless.

3. Some Flaws of the Previous Works on the UKS Attacks

3.1 DSKS-UKS Attack on the Revised STS-MAC

To prevent DSKS-UKS attacks on STS-MAC protocol, Blake-Wilson and Menezes recommended including identities of A and B and flow numbers in the messages being signed. Applying this, they proposed a revised version of STS-MAC protocol as shown in Fig 2.

(1b)	$A \to B$	I_A, α^{r_A}
(2b)	$A \leftarrow B$	$Cert_B, \alpha^{r_B}, S_B(2, \alpha^{r_B}, \alpha^{r_A}, I_B, I_A)$, $MAC_K(S_B(2, \alpha^{r_B}, \alpha^{r_A}, I_B, I_A))$
(3b)	$A \to B$	$Cert_A, S_A(3, \alpha^{r_A}, \alpha^{r_B}, I_A, I_B)$, $MAC_K(S_A(3, \alpha^{r_A}, \alpha^{r_B}, I_A, I_B))$

Fig. 2 Revised STS-MAC Protocol.

However, we show that this protocol is still vulnerable to a DSKS-UKS attack. Suppose that the signature scheme in the revised STS-MAC protocol is ElGamal one (We confirmed that the same attack can be launched in the case of the RSA.) and an adversary E for this UKS attack against the responder acts as described in Section 2. Namely, E intercepts the message in flow (1b) and replace it with (I_E, α^{r_A}). Similarly, E intercepts the message in flow (3b) and replace it with $(Cert_E, S_A(3, \alpha^{r_A}, \alpha^{r_B}, I_A, I_B)$, $MAC_K(S_A(3, \alpha^{r_A}, \alpha^{r_B}, I_A, I_B)))$. Even though identities of A and B as well as flow numbers are included in the message being signed, B verifies the signature in the message in flow (3b) with information contained in the certificate of E as shown in the following proposition.

Proposition 1: Assume that the signature scheme in the revised STS-MAC protocol is ElGamal one. Let $(p, g, y(= g^a))$ be public-key parameters of the ElGamal scheme, where g is a generator for a given multiplicative group Z_p^* and a is kept as secret. Suppose that the signature on $(3, \alpha^{r_A}, \alpha^{r_B}, I_A, I_B)$ is $(r, s) = (g^k, k^{-1}(h(3, \alpha^{r_A}, \alpha^{r_B}, I_A, I_B) - ar))$. Then E can select her own public-key of the ElGamal signature scheme for B to verify the signature (r, s) as being correct.

Proof: First, E computes $h' = h(3, \alpha^{r_A}, \alpha^{r_B}, I_E, I_B)$. Note here that I_E is included in the message being hashed instead of I_A. Let q be a large prime number which divides $p-1$. If $\gcd(s, p-1) \neq 1$ or if $\gcd(h', r) = 2$ or q, then E terminates with failure. Otherwise, E selects $a' \in Z_q$ at random. Then $t = h' - a'r$ is computed and the new public parameter $(p, \bar{g}, \bar{y})$ is set to $(p, (r^s)^{t^{-1}}, \bar{g}^{a'})$.

B then verifies the signature (r, s) with the public parameter
$(p, \bar{g}, \bar{y})$. After computing $h' = h(3, \alpha^{r_A}, \alpha^{r_B}, I_E, I_B)$, he checks whether $\bar{g}^{(-h')}\bar{y}^r r^s \bmod p$ is 1. If the result is 1, the signature is verified as being correct. Note that it actually yields 1 since

$$\begin{aligned} & \bar{g}^{(-h')}\bar{y}^r r^s \bmod p \\ = & (r^{st^{-1}})^{(-h')}(r^{st^{-1}})^{a'r} r^s \\ = & (r^{s/(h'-a'r)})^{(-h')}(r^{s/(h'-a'r)})^{a'r} r^s \\ = & r^{(-sh'+sa'r+sh'-sa'r)/(h'-a'r)} \\ = & r^0 \\ = & 1. \end{aligned}$$

□

Remarks: The condition $\gcd(s, p-1) = 1$ guarantees that r^s, *i.e.*, $\bar{g}$ is a generator of Z_p^*. Also, the condition $\gcd(h', r) \neq 2$ or q guarantees that there exists a' such that $\gcd(t, p-1) = 1$.

According to the result of Proposition 1, including identities and flow numbers in the messages being signed is not sufficient to prevent DSKS-UKS attacks.

Therefore, the key agreement protocols employing signature schemes in [3], [5], and [10] can be vulnerable to DSKS-UKS attacks *provided that* each entity's certificate of a public key for signature verification is not exchanged before the protocol begins.

3.2 Public Key Substitution UKS Attacks on KAP-HY98

At PKC'98, Hirose and Yoshida [8] proposed a new key agreement protocol, which we call "KAP-HY98." It was shown that KAP-HY98 is secure against passive eavesdropping, impersonation, interference, active eavesdropping, and pretense in the random oracle model. The proposed signature scheme for providing authentication in KAP-HY98 is called a redundant signature scheme as described below in brief.

Redundant Signature Scheme

Let $S_B \in \mathrm{Z}_q$ be a secret of entity 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 $e_B = h(z_B, u_B, u_A)$.

KAP-HY98 was designed based on the redundant signature scheme. Note that, in [8], the public key for signature verification is assumed to be contained in identifying information of each party, 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 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 the redundant signature scheme. B sends (u_B, e_B, w_B) and I_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, 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, B terminates the protocol. Otherwise, computes $K_B = u_A^{-k_B} \bmod p$, which becomes the session key between A and B.

Now we show that KAP-HY98 is vulnerable to public key substitution UKS attacks.

As shown in Fig. 3, a public key substitution attack against the initiator can be launched. An adversary E registers B's public key P_B as its own; that is, $P_E = P_B$. In the first stage of this attack, E simply passes the message from A to B unchanged in flows (1c) and (1'c). When B sends A the message in flow (2c), E intercepts it, changes the certificate of P_B into that of P_E, and forwards this changed message to A in flow (2'c). After receiving (e_B, w_B) from E, A verifies (e_B, w_B) as being correct since any identity is not included in the message being signed and $P_E = P_B$. E then passes the message in flow (3c) from A to B unchanged. At the end of this attack, A falsely believes that the common session key is shared with E, while B believes that the key is shared with A.

(1c)	$A \rightarrow E$	u_A, I_A
(1'c)	$E \rightarrow B$	u_A, I_A
(2c)	$A \hookleftarrow B$	$(u_B, e_B, w_B), I_B$
(2'c)	$A \leftarrow E$	$(u_B, e_B, w_B), I_E$
(3c)	$A \rightarrow E$	e_A, w_A
(3'c)	$E \rightarrow B$	e_A, w_A

Fig. 3 Public Key Substitution UKS Attack against the Initiator on KAP-HY98.

Also, KAP-HY98 is vulnerable to a public key substitution attack against the responder. Let us examine the following public key substitution UKS attack against the responder.

(1d)	$A \hookrightarrow E$	u_A, I_A
(1'd)	$E \rightarrow B$	u_A, I_E
(2d)	$A \leftarrow B$	$(u_B, e_B, w_B), I_B$
(2'd)	$A \leftarrow E$	$(u_B, e_B, w_B), I_B$
(3d)	$A \rightarrow E$	e_A, w_A
(3'd)	$E \rightarrow B$	e_A, w_A

Fig. 4 Public Key Substitution UKS Attack against the Responder on KAP-HY98.

In this attack, an adversary E registers A's public key P_A as its own; that is, $P_E = P_A$. When A sends B the message in flow (1d), E intercepts it and changes the certificate of P_B into that of P_E in flow (1'd). E then passes the message from B to A unchanged in flows (2d) and (2'd). Finally, E also passes the message from A to B unchanged in flows (3d) and (3'd). Since no identity is included in (e_A, w_A) and $P_E = P_A$, the signed message (e_A, w_A) is verified as being correct by B.

We close this section by stating that a DSKS-UKS attack on the KAP-HY98 seems to be impossible since the same generator g is used for both session key generation and the redundant signature scheme, which means that g must be shared among entities before the protocol execution.

4. Prevention of UKS Attacks

There are some proposed methods for preventing UKS attacks:

(A) Certificates of both entities should be exchanged prior to the key agreement protocol.

(B) CA should issue certificates with checking that each entity possesses a private key corresponding to its public key.

(C) Identities of the sender and intended receiver as well as flow numbers should be included in the messages being signed.

Although **(A)** is often assumed in a number of authenticated key agreement protocol schemes, it is not desirable in situations that require low bandwidth. **(B)** is a desired property of CA service but it is controversial whether every CA will conduct such checking procedures. Moreover, if employed signature schemes for key agreement protocols have the DSKS property, we emphasize that it is of no use for CA to check possession of private keys. Hence, **(C)** is considered to be a better method since there will be no additional overhead on run of key agreement protocols. However, care must be taken when applying this method, which is well applied to public key substitution UKS attacks but *not always applied to prevent DSKS-UKS attacks.* Consequently, we propose another method to prevent DSKS-UKS attacks:

(D) Identities of the sender and intended receiver as well as flow numbers should be included in both the messages being MACed and the messages being signed if MAC is employed in the key agreement protocol.

From **(D)**, we can modify the revised STS-MAC protocol as shown in Fig. 5.

$$\begin{array}{lll}
(1e) & A \rightarrow B & I_A, \alpha^{r_A} \\
(2e) & A \leftarrow B & Cert_B, \alpha^{r_B}, S_B(2, \alpha^{r_B}, \alpha^{r_A}, I_B, I_A), \\
 & & MAC_K(S_B(2, \alpha^{r_B}, \alpha^{r_A}, I_B, I_A), 2, I_B, I_A) \\
(3e) & A \rightarrow B & Cert_A, S_A(3, \alpha^{r_A}, \alpha^{r_B}, I_A, I_B), \\
 & & MAC_K(S_A(3, \alpha^{r_A}, \alpha^{r_B}, I_A, I_B), 3, I_A, I_B)
\end{array}$$

Fig. 5 Modified STS-MAC Protocol.

Since an identity of each entity is included in the messages being MACed, modification of this identity will be detected by each entity even though an adversary has selected a duplicate signature key and changed one of the identities. Note that the key agreement mechanism 7 in ISO/IEC 11770-3 [9], where identities of involving parties are directly MACed, is secure against DSKS-UKS attacks.

5. Concluding Remarks

Another aspect of significance of UKS attacks has been presented by pointing out some flaws in previous works on UKS attacks. We have shown that the revised STS-MAC protocol proposed by Blake-Wilson and Menezes is still vulnerable to a DSKS-UKS attack and KAP-HY98 is insecure against public key substitution UKS attacks. Therefore, we conclude that public key certificates for signature schemes in key agreement protocols should be carefully handled. We finally remark that the recent Shoup's formal models [13] for key agreement protocols are dealing with this problem with great emphasis.

References

[1] ANSI X9.30 (Part 1), "Public key cryptography using irrevisible algorithms for the financial service industry - Part 1: The digital signature algorithm (DSA)", 1995.

[2] ANSI X9.62, "The elliptic curve digital signature algorithm (ECDSA)", Working Draft, March 1995.

[3] M. Bellare, R. Canetti, and H. Krawczyk, "A modular approach to the design and analysis of authentication and key exchange protocols", Proc. of Annual Symposium on the Theory of Computing, 1998, (`http://www-cse.ucsd.edu/users/mihir`).

[4] S. Blake-Wilson and A. Menezes, "Unknown key-share attacks on the Station-to-Station (STS) protocol", Proc. of PKC'99, LNCS 1560, pp.154–170, Springer-Verlag, 1999.

[5] S. Blake-Wilson and A.Menezes, "Entity authentication and authenticated key agreement protocols employing asymmetric techniques", Proc. of Security Protocols Workshop'97, LNCS 1361, pp.137–158, Springer-Verlag, 1997.

[6] W. Diffie, P. van Oorschot and M. Wiener, "Authentication and authenticated key exchanges", Designs, Codes and Cryptography, Vol.2, pp.107–125, 1992.

[7] 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.

[8] S. Hirose and S. Yoshida, "An authenticated Diffie-Hellman key agreement protocol secure against active attacks", Proc. of PKC'98, LNCS 1431, pp.135–148, Springer-Verlag, 1998.

[9] ISO/IEC 11770-3, "Information technology-security techniques-key management - Part 3: Mechanisms using asymmetric techniques", Final Text for Publication, 1999.

[10] C. Mitchell and A. Thomas, "Standardising authentication protocols based on public key techniques", Journal of Computer Security, Vol. 2, pp.23–46, 1993.

[11] M. O. Rabin,"Digitalized signatures and public-key functions as intractable as factorization", MIT/LCS/TR-212, MIT Laboratory for Computer Science, 1979.

[12] R. L. Rivest, A. Shamir, L. M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", Communications of the ACM, Vol.21, pp.120–126, 1978.

[13] V. Shoup, "On formal models for secure key exchange", IBM Research Report RZ 3120, 1999, (`http://www.shoup.net/papers`)