Strong Diffie-Hellman-DSA Key Exchange

Ik Rae Jeong, Jeong Ok Kwon, and Dong Hoon Lee

***Abstract*— To provide authentication to the Diffie-Hellman key exchange, a few integrated key exchange schemes which provide authentication using the DSA signature have been proposed in the literature. In this letter we point out that all of the previous Diffie-Hellman-DSA schemes do not provide security against *session state reveal* attacks. We also suggest a strong Diffie-Hellman-DSA scheme providing security against session state reveal attacks as well as *forward secrecy* and *key independence*.**

***Index Terms*— Cryptography, Diffie-Hellman key exchange, digital signature algorithm.**

I. Introduction

TO provide authentication to the Diffie-Hellman key exchange [3], Arazi used DSA (Digital Signature Algorithm) [1]. Unfortunately the integrated key exchange scheme of [1] does not provide *key independence* [8]. Harn *et al.* modified the scheme of [1] to provide key independence [5]. But the scheme in [5] does not provide *forward secrecy* [9]. Phan modified the scheme of [5] to provide forward secrecy [9].

The security against *session state reveal* is formally considered in [2], [6]. This security is originated from the consideration that the ephemeral random values of the sessions may be more easily leaked than the secret keys of the public keys.

In this letter, we show that all of the schemes suggested in [5], [9] are insecure against session state reveal attacks. We then suggest a strong Diffie-Hellman-DSA scheme which provides security against session state reveal attacks as well as forward secrecy and key independence.

II. DSA Signature Scheme [7]

The DSA signature scheme consists of $\mathsf{DSA} = (\mathsf{DSA.key}, \mathsf{DSA.gen}, \mathsf{DSA.ver})$. DSA.key generates a private-public key pair for the party. That is, a public key consists of a prime p, an order q which is also a prime, a generator g, and y. A private key is x such that $y = g^x \mod p$. DSA.gen makes a signature (r, s) for a message m with the private key x such that $r = ((g^k \mod p) \mod q)$ and $s = (k^{-1} \cdot (H(m) + xr)) \mod q$, where k is a random value and H is a hash function. DSA.ver verifies a message-signature pair (m, r, s) with the public key y and returns 1 if valid or 0 otherwise. That is, the algorithm checks $0 < r, s < q$ and $((g^{H(m)s^{-1}} y^{rs^{-1}}) \mod p) \mod q \stackrel{?}{=} r$.

Manuscript received January 2, 2006. The associate editor coordinating the review of this letter and approving it for publication was Prof. Hsiao-Hwa Chen.

I. R. Jeong is with ETRI (Electronics and Telecommunications Research Institute), Daejeon, Korea (e-mail: jir@etri.re.kr).

J. O. Kwon and D. H. Lee are with the Graduate School of Information Security CIST, Korea University, Seoul, Korea (e-mail: {pitapat, donghlee}@korea.ac.kr). J. O. Kwon and D. H. Lee were supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Advancement) (IITA-2006-(C1090-0603-0025)).

Digital Object Identifier 10.1109/LCOMM.2007.070004.

III. Notions of Key Exchange

We review the most widely used security notions of the key exchange schemes in the literature.

Key Independence. This is a stronger notion of security and means that session keys are computationally independent from each other. A bit more formally, key independence protects against "Denning-Sacco" attacks [4] involving compromise of multiple session keys (for sessions other than the one whose secrecy must be guaranteed).

Forward Secrecy. The protocols achieving forward secrecy maintain the secrecy of session keys even when an adversary is able to obtain long-term secret keys of principals who have generated session keys.

Security against Session State Reveal. The protocols providing security against session state reveal attacks maintain the secrecy of session keys even when an adversary is able to obtain the random numbers used to make the session keys.

IV. Insecurity of HMH [5] and P [9] Schemes

In [5], Harn *et al.* suggest three key exchange protocols. We refer the third protocol of [5] as HMH. In [9], Phan suggests a key exchange protocol which is a modified version of HMH. We refer the modified version of HMH as P. P is depicted in Fig 1. We denote the concatenation of two strings a and b as $a||b$.

In the protocols the two session keys, K_{AB} and K_{BA}, are made in a session. K_{AB} may be used as a cryptographic key for a communication from user A to user B, and K_{BA} may be used as a cryptographic key for a communication from user B to user A.

Theorem 1. P and HMH are insecure against session state reveal attacks.

Proof of Theorem 1. If an adversary $\mathcal{A}$ gets the random numbers used by user A and user B, $\mathcal{A}$ can calculate the session keys, K_{AB} and K_{BA}. In P, K_{AB} and K_{BA} are calculated as $K_{AB} = g^{x_B vw} \mod p$ and $K_{BA} = g^{x_A vw} \mod p$, where v and w are random numbers selected by user A and B. If $\mathcal{A}$ gets v and w, $\mathcal{A}$ can easily calculate $K_{AB} = y_B^{vw} \mod p$ and $K_{BA} = y_A^{vw} \mod p$. Thus, P is insecure against session state reveal attacks.

User A (x_A, y_B)		User B (x_B, y_A)
Select a random integer v $m_A = g^v \mod p$, $n_A = y_A^v \mod p$	$\xrightarrow{m_A, n_A}$	Select a random integer w $K_{BA} = n_A^w \mod p = g^{x_A v w} \mod p$ $K_{AB} = m_A^{x_B w} \mod p = g^{x_B v w} \mod p$ $m_B = g^w \mod p$, $n_B = y_B^w \mod p$ $r_B = m_B \mod q$
$K_{AB} = n_B^v \mod p = g^{x_B v w} \mod p$ $K_{BA} = m_B^{x_A v} \mod p = g^{x_A v w} \mod p$ $r_B = m_B \mod q$, $r_A = m_A \mod q$ Check $\mathsf{DSA.ver}_{y_B}(m_B \| K_{BA} \| K_{AB}, r_B, s_B) \stackrel{?}{=} 1$	$\xleftarrow{m_B, n_B, s_B}$	$s_B = (w^{-1} \cdot (H(m_B \| K_{BA} \| K_{AB}) + x_B r_B)) \mod q$
$s_A = (v^{-1} \cdot (H(m_A \| K_{AB} \| K_{BA}) + x_A r_A)) \mod q$	$\xrightarrow{s_A}$	$r_A = m_A \mod q$ Check $\mathsf{DSA.ver}_{y_A}(m_A \| K_{AB} \| K_{BA}, r_A, s_A) \stackrel{?}{=} 1$

Fig. 1. P protocol [9]

User A (x_A, y_B)		User B (x_B, y_A)
Select a random integer v $m_A = g^v \mod p$	$\xrightarrow{m_A}$	Select a random integer w $DH1 = m_A^w \mod p = g^{vw} \mod p$ $DH2 = y_A^{x_B} \mod p = g^{x_A x_B} \mod p$ $m_B = g^w \mod p$, $r_B = m_B \mod q$
$DH1 = m_B^v \mod p = g^{vw} \mod p$ $DH2 = y_B^{x_A} \mod p = g^{x_A x_B} \mod p$ $r_B = m_B \mod q$, $r_A = m_A \mod q$ Check $\mathsf{DSA.ver}_{y_B}(m_B \| DH1 \| DH2, r_B, s_B) \stackrel{?}{=} 1$ $K_{AB} = H(A \| B \| DH1 \| DH2)$ $K_{BA} = H(B \| A \| DH1 \| DH2)$	$\xleftarrow{m_B, s_B}$	$s_B = (w^{-1} \cdot (H(m_B \| DH1 \| DH2) + x_B r_B)) \mod q$
$s_A = (v^{-1} \cdot (H(m_A \| DH1 \| DH1) + x_A r_A)) \mod q$	$\xrightarrow{s_A}$	$r_A = m_A \mod q$ Check $\mathsf{DSA.ver}_{y_A}(m_A \| DH1 \| DH2, r_A, s_A) \stackrel{?}{=} 1$ $K_{AB} = H(A \| B \| DH1 \| DH2)$ $K_{BA} = H(B \| A \| DH1 \| DH2)$

Fig. 2. The suggested strong Diffie-Hellman-DSA protocol

In HMH, K_{AB} and K_{BA} are calculated as $K_{AB} = g^{x_B v} \mod p$ and $K_{BA} = g^{x_A w} \mod p$, where v and w are random numbers selected by user A and B. If $\mathcal{A}$ gets v and w, $\mathcal{A}$ can easily calculate $K_{AB} = y_B^v \mod p$ and $K_{BA} = y_A^w \mod p$. Thus, HMH is insecure against session state reveal attacks. □

V. Our Strong Diffie-Hellman-DSA Scheme

We suggest a key exchange scheme based on the DSA signature scheme. Our scheme provides security against session state reveal attacks and forward secrecy as shown in Fig 2.

The basic idea of our protocol is as follows. In our scheme, the session keys are $K_{AB} = H(A \| B \| g^{vw} \| g^{x_A x_B})$ and $K_{BA} = H(B \| A \| g^{vw} \| g^{x_A x_B})$. Even if an adversary knows v and w, the adversary cannot calculate $g^{x_A x_B}$. So our protocol provides security against session state reveal attacks. Even if an adversary knows x_A and x_B, the adversary cannot calculate g^{vw}. So our protocol provides forward secrecy. In each session, the new ephemeral random numbers are used, so an adversary cannot know the session keys of the intact sessions even with some session keys of the other sessions. Thus, our protocol provides key independence.

VI. Conclusion

We have showed that the DSA-based Diffie-Hellman key exchange schemes in [5], [9] are not secure against the session state reveal attacks. We have then suggested a strong Diffie-Hellman-DSA key exchange scheme providing security against session state attacks as well as forward secrecy and key independence. Our suggested Diffie-Hellman-DSA key exchange scheme is more efficient than the scheme in [9]. That is, our scheme requires one less exponentiations than the scheme in [9].

References

[1] A. Arazi, "Integrating a key cryptosystem into the digital signature standard," *Electron. Lett.*, vol. 29, pp. 966-967, Nov. 1993.

[2] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *Advances in Cryptology-EUROCRYPT 2001*, pp. 453-474, Springer Verlag, 2001.

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

[4] D. Denning and G. M. Sacco, "Timestamps in key distribution protocols," *Commun. ACM*, vol. 24, no. 8, pp. 533–536, 1981.

[5] L. Harn, M. Mehta, and W.-J. Hsin, "Integrating Diffie-Hellman key exchange into the digital signature algorithm (DSA)," *IEEE Commun. Lett.*, vol. 8, pp. 198-200, Mar. 2004.

[6] H. Krawczyk, "HMQV: a high-performance secure Diffie-Hellman Protocol," in *Proc. CRYPTO'05*, pp. 546-566.

[7] National Institute of Standards and Technology, Digital Signature Standard (DSS), *Federal Information Processing Standards Publication*, FIPS PUB 186-2, Reaffirmed, Jan. 27, 2000.

[8] K. Nyberg and R. A. Rueppel, "Weaknesses in some recent key agreement protocols," *Electron. Lett.*, vol. 30, pp. 26-27, Jan. 1994.

[9] R. C.-W. Phan, "Fixing the integrated Diffie-Hellman-DSA key exchange protocol," *IEEE Commun. Lett.*, vol. 9, pp. 570-572, June 2005.