References

1 SHAPIRO, J.M.: 'Embedded image coding using zerotrees of wavelet coefficients', *IEEE Trans. Signal Process.*, 1993, **41**, pp. 3445–3462

2 SAID, A., and PEARLMAN, W.A.: 'A new fast and efficient image codec based on set partitioning in hierarchical trees', *IEEE Trans. Circuits Syst. Video Technol.*, 1996, **6**, pp. 1–15

3 CHAI, BING-BING, VASS, J., and ZHUANG, XINHUA: 'Significance-linked connected component analysis for wavelet image coding', *IEEE Trans. Image Process.*, 1999, **8**, (6), pp. 774–784

4 XIONG, ZIXIANG, RAMCHANDRAN, K., and ORCHARD, M.T.: 'Space-frequency quantization for wavelet image coding', *IEEE Trans. Image Process.*, 1997, **6**, (5), pp. 677–693

Authenticated key agreement without using one-way hash functions

L. Harn and H.-Y. Lin

The MQV key agreement protocol has been adopted by the IEEE P1363 Committee to become a standard. The MQV protocol used a digital signature to sign the Diffie-Hellman public keys without using any one-way function. Here, the MQV protocol is generalised in three respects. First, signature variants for Diffie-Hellman public keys developed previously are employed in the new protocol. Secondly, two communication entities are allowed to establish multiple secret keys in a single round of message exchange. Thirdly, the key computations are simplified.

Introduction: Diffie and Hellman [1] proposed in 1976 the well-known public-key distribution scheme based on the discrete logarithm problem to enable two parties to establish a common secret session key based on their exchanged public keys; however, their original scheme still requires an authentication channel to exchange the public keys. Since then, several key exchange protocols [2, 3], which use digital signatures of the exchanged public keys to provide authentication, have been proposed. In these protocols, the Diffie-Hellman public keys are treated as messages and one-way hash values of these public keys are computed. Digital signatures need to be generated based on these one-way hash values; otherwise forgery can be easily achieved.

However, there exists a major difference of security assumptions between digital signature schemes and conventional one-way hash functions. The security assumption of most signature schemes are based on some well-known computational problems, such as the discrete logarithm problem [4] and the factoring problem. The complexities of these problems have been well studied and the difficulties of solving them are recognised. In contrast, the security of a one-way hash function is based on the complexity of analysing a simple iterated function. A one-way hash function may seem very difficult to analyse at the beginning, but it may turn out to be vulnerable to some special attacks later, e.g. recent advancement in cryptanalytic research has found that MD5 is at the edge of risking successful cryptanalytic attack [5]. Instead of overall security relying on the weaker assumption of the signature scheme and the one-way hash function, it would be more secure to have a key distribution without using one-way hash functions.

The MQV key agreement protocol proposed by Menezes *et al.* [6] in 1995 is probably the first key agreement protocol that utilised a signature for the Diffie-Hellman public key without using a one-way hash function. The MQV key agreement protocol has been adopted to become a standard in the IEEE P1363 committee [7]. In 1998, we published a key agreement protocol [8] that generalised the MQV protocol in three respects. First, signature variants for Diffie-Hellman public keys developed in 1997 [9] are employed in the protocol. Secondly, we allowed two communication entities to establish multiple secret keys in one round of interaction. Thirdly, we simplified the key computations.

Two attacks [10, 11] on this key agreement protocol were found recently. In this Letter, we attempt to show that these two attacks can easily be avoided by modifying the signature signing equation. We point out here that this modification does not increase computations. The main body of this Letter is almost the same as [8] except that we have modified the signature signing and verification equations.

Digital signature schemes for Diffie-Hellman public keys: The Diffie-Hellman public key is $r = \alpha^k \bmod p$, where k is a secret random integer within $[1, p - 2]$ privately selected by the signer, p is a large prime and α is a primitive number in $GF(p)$. We can call these random parameters, k and r, short-term private key and short-term public key, respectively. To sign this Diffie-Hellman public key r, the signer needs to use the long-term private key x and the long-term public key $y = \alpha^x \bmod p$.

We list four signature variants for signing Diffie-Hellman public keys from [9]. These variants are the most efficient variants since each permutes four parameters $\{r, s, k, x\}$ directly. The signer needs to use its short-term and long-term secret keys, k and x, to generate the signature s for the Diffie-Hellman public key r. However, the verifier can use the signer's long-term public key y to verify the signature s for r. (See [9] for detailed discussions.) We point out that the MQV key agreement protocol in [7] actually used one of the variants in Table 1.

Table 1: Signature variants

Signature equation	Signature verification
$rx = k + s \bmod p - 1$	$y^r = r\alpha^s \bmod p$
$sx = k + r \bmod p - 1$	$y^s = r\alpha^r \bmod p$
$x = rk + s \bmod p - 1$	$y = r^r\alpha^s \bmod p$
$x = sk + r \bmod p - 1$	$y = r^s\alpha^r \bmod p$

Key agreement protocols: In the following discussion we assume that A and B want to establish a secret key(s) using the key agreement protocol. The short-term secret key and short-term public key for A are k_A and r_A, and the long-term secret key and long-term public key for A are x_A and y_A. Similarly, B has k_B, r_B, x_B and y_B. The following key agreement protocol enables A and B to establish an authenticated secret key K.

(i) A generates a random short-term secret key k_A and its corresponding public key r_A, and computes the signature s_A based on any variant as listed in Table 1. A sends $\{r_A, s_A, cert(y_A)\}$ to B, where $cert(y_A)$ is the public-key certificate of y_A signed by a trusted party.

(ii) Similarly, B generates k_B, r_B, s_B and sends $\{r_B, s_B, \mathrm{cert}(y_B)\}$ to A.

(iii) A verifies r_B based on the signature s_B and B's public key y_B. Then A computes the shared secret key $K = r_B^{k_A} \bmod p$.

(iv) Similarly, B verifies r_A based on the signature s_A and A's public key y_A. Then B computes the shared secret key $K = r_A^{k_B} \bmod p$.

Possible attacks: One drawback of the above protocol is that it does not offer perfect forward secrecy [12], i.e. if an adversary learns one shared secret key they can deduce all shared secret keys between A and B. We illustrate this attack in the following discussion.

Assume that the protocol uses $x = rk + s \bmod p - 1$ to sign each Diffie-Hellman public key. We then have the following two equations:

$$x_A = r_A k_A + s_A \bmod p-1 \text{ and } x_B = r_B k_B + s_B \bmod p-1$$

By multiplying these two equations, we obtain

$$x_A x_B = r_A k_A r_B k_B + r_A k_A s_B + s_A r_B k_B + s_A s_B \bmod p-1$$

In other words,

$$K_{AB} = (K^{r_A r_B})(r_A^{r_A s_B})(r_B^{r_B s_A})(\alpha^{s_A s_B}) \bmod p$$

where $K_{AB} = \alpha^{x_A x_B} \bmod p$ is the long-term shared secret key. Assume that the adversary knows one short-term shared secret key K. The adversary can then solve the long-term shared secret key K_{AB} from the above equation since the other parameters are all known. Thus, the adversary can solve all other shared secret keys based on the same equation.

Improved protocol: In the two-pass MQV key agreement protocol, instead of using $K = \alpha^{k_A k_B} = r_B^{k_A} = r_A^{k_B} \bmod p$ as the shared secret key, it uses $K = \alpha^{x_A k_B + x_B k_A} = y_A^{k_B} r_A^{x_B} = y_B^{k_A} r_B^{x_A} \bmod p$ as the shared secret key. The MQV protocol can provide perfect forward secrecy.

Here, we want to propose an efficient protocol that enables A and B to share multiple secret keys in one round of message exchange. For simplicity, we assume that A and B want to share four secrets.

(i) A generates two random short-term secret keys, k_{A_1} and k_{A_2}, and two corresponding public keys, r_{A_1} and r_{A_2}, $r_{A_1} < r_{A_2}$. Then, A computes the signature s_A for $\{r_{A_1}, r_{A_2}\}$ based on any signature variant as listed in Table 1. For example, A obtains s_A by solving the following equation

$$x_A = r_{A_1} k_{A_1} + r_{A_2} k_{A_2} + s_A \bmod p - 1$$

A sends $\{r_{A_1}, r_{A_2}, s_A, cert(y_A)\}$ to B, where $cert(y_A)$ is the public-key certificate of y_A signed by a trusted party.

(ii) Similarly, B generates k_{B_1}, k_{B_2}, r_{B_1}, r_{B_2}, s_B and sends $\{r_{B_1}, r_{B_2}, s_B, cert(y_B)\}$ to A.

(iii) A verifies $\{r_{B_1}, r_{B_2}\}$ based on the signature s_B and B's public key y_B by checking

$$y_B = r_{B_1}^{r_{B_1}} r_{B_2}^{r_{B_2}} \alpha^{s_B} \bmod p$$

Then A computes the shared secret keys as

$$K_1 = r_{B_1}^{k} A_1 \bmod p$$
$$K_2 = r_{B_1}^{k} A_2 \bmod p$$
$$K_3 = r_{B_2}^{k} A_1 \bmod p$$
$$K_4 = r_{B_2}^{k} A_2 \bmod p$$

(iv) Similarly, B computes $\alpha^{r_{A_1} r_{A_2}} \bmod p$ first and verifies $\{r_{A_1}, r_{A_2}\}$. Then, B computes the shared secret keys as

$$K_1 = r_{A_1}^{k} B_1 \bmod p$$
$$K_2 = r_{A_2}^{k} B_1 \bmod p$$
$$K_3 = r_{A_1}^{k} B_2 \bmod p$$
$$K_4 = r_{A_2}^{k} B_2 \bmod p$$

Discussion: We point out here that we have modified the original protocol [8] in signature signing and verification equations. Two recent attacks [10, 11] on the original protocol cannot work successfully in this modified protocol. This modified protocol does not increase any computational load and the key agreement protocol does not involve any additional one-way hash function.

The signatures, x_A and x_B, satisfy the following equations as

$$x_A = r_{A_1} k_{A_1} + r_{A_2} k_{A_2} + s_A \bmod p - 1 \quad \text{and}$$
$$x_B = r_{B_1} k_{B_1} + r_{B_2} k_{B_2} + s_B \bmod p - 1$$

By multiplying these two equations together, we obtain

$$\begin{aligned} x_A x_B = {} & r_{A_1} r_{B_1} k_{A_1} k_{B_1} + r_{A_1} r_{B_2} k_{A_1} k_{B_2} + r_{A_1} s_B k_{A_1} \\ & + r_{A_2} r_{B_1} k_{A_2} k_{B_1} + r_{A_2} r_{B_2} k_{A_2} k_{B_2} + r_{A_2} s_B k_{A_2} \\ & + s_A r_{B_1} k_{B_1} + s_A r_{B_2} k_{B_2} + s_A s_B \bmod p - 1 \end{aligned}$$

In other words, we have

$$\begin{aligned} K_{AB} = {} & K_1^{r_{A_1} r_{B_1}} K_2^{r_{A_2} r_{B_1}} K_3^{r_{A_1} r_{B_2}} K_4^{r_{A_2} r_{B_2}} \\ & \times r_{A_1}^{r_{A_1} s_B} r_{A_2}^{r_{A_2} s_B} r_{B_1}^{r_{B_1} s_A} r_{B_2}^{r_{B_2} s_A} \alpha^{s_A s_B} \bmod p \end{aligned}$$

If the adversary knows four consecutive shared secret keys, he can solve the long-term shared secret K_{AB}. Thus, to achieve the perfect forward secrecy, we should limit ourselves to use only three out of the four shared secret keys. The protocol can be generalised to enable A and B to share $n^2 - 1$ secrets if each user computes and sends n Diffie-Hellman public keys in each pass. Since each user only needs to generate (verify) one signature for n different Diffie-Hellman public keys to establish $n^2 - 1$ shared secret keys, this new protocol is very efficient.

Conclusion: We have proposed an authenticated key agreement protocol that utilises a digital signature to authenticate Diffie-Hellman public keys. We summarise features in this new protocol as follows:

(i) Since we integrate the Diffie-Hellman public key in the signature scheme, this approach reduces overall computation.

(ii) Since the protocol does not use any one-way hash function, the security assumption relies solely on solving the discrete logarithm problem.

(iii) This protocol allows two communication parties to share multiple secret keys in two-pass interaction.

(iv) The computation for shared secret keys is simpler than the MQV protocol.

14 March 2001
Electronics Letters Online No: 20010441
DOI: 10.1049/el:20010441

L. Harn (*Department of Computer Networking, University of Missouri, Kansas City, MO 64110, USA*)

H.-Y. Lin (*Computer Science Department, California State University, San Marcos, CA 92096-0001, USA*)

References

1 DIFFIE, W., and HELLMAN, M.E.: 'New directions in cryptography', *IEEE Trans. Inf. Theory,* 1976, **IT-22**, (6), pp. 644–654
2 ARAZI, A.: 'Integrating a key cryptosystem into the digital signature standard', *Electron. Lett.*, 1993, **29**, (11), pp. 966–967
3 NYBERG, K., and RUEPPEL, R.A.: 'Message recovery for signature scheme based on the discrete logarithm problem'. Proc. Eurocrypt '94, May 1994, pp. 175–190
4 ELGAMAL, T.: 'A public-key cryptosystem and a signature scheme based on discrete logarithms', *IEEE Trans. Inf. Theory,* 1985, **IT-31**, pp. 469–472
5 DOBBERTIN, H.: 'The status of MD5 after a recent attack', *CryptoBytes,* 1996, **2**, (2), pp. 1–6
6 MENEZES, A.J., QU, M., and VANSTONE, S.A.: 'Some key agreement protocols providing implicit authentication'. 2nd Workshop Selected Areas in Cryptography, 1995
7 IEEE P1363/Editorial Contribution (Draft). In http://stdsbbs.ieee.org/groups/1363/edcont.html
8 HARN, L., and LIN, H.Y.: 'An authenticated key agreement protocol without using one-way function'. Proc. 8th Nat. Conf. Information Security, Kaohsiung, Taiwan, May 1998, pp. 155–160
9 HARN, L.: 'Digital signatures for Diffie-Hellman public keys without using one-way function', *Electron. Lett.*, 1997, **33**, (2), pp. 125–126
10 YEN, S.M., and JOYE, M.: 'Improved authenticated multiple-key agreement protocol', *Electron. Lett.*, 1998, **34**, (18), pp. 1738–1739
11 WU, T.S., HE, W.H., and HSU, C.L.: 'Security of authenticated multiple-key agreement protocols', *Electron. Lett.*, 1999, **35**, (5), pp. 391–392
12 LIM, C.H., and LEE, P.J.: 'Security of interactive DSA batch verification', *Electron. Lett.*, 1994, **30**, (19), pp. 1592–1593

Cryptanalysis on improved user efficient blind signatures

C.-I. Fan and C.-L. Lei

Shao proposed a blind signature scheme based on the Fan-Lei scheme. It is shown here that Shao's scheme is not secure. Also, Shao claimed that the Fan-Lei scheme is not really blind, however this claim is demonstrated as not being true.

Introduction: In 1996, Fan and Lei proposed a blind signature scheme based on quadratic residues (QRs) [1], and they also presented an enhanced version of the scheme to reduce the computation for requesters or users [2]. In [3], Shao proposed a blind signature scheme based on the Fan-Lei scheme [2]. However, we find that Shao's scheme cannot withstand Pollard-Schnorr attacks [4]. Besides, Shao claimed that the Fan-Lei blind signature scheme [2] is not really blind. In this Letter, we also show that Shao's claim is not true.

Attacks on Shao's blind signature scheme: Shao proposed a blind signature scheme based on the Fan-Lei scheme in [3]. We show that Pollard-Schnorr attacks [4] are valid on Shao's scheme as follows. In the scheme of [3], the tuple (c, s) is a signature of m and they can be verified by checking if

$$H(m) s^2 (c^2 + 1) = 1 \bmod n \qquad (1)$$

An attacker can choose a message m and then derive (w, y) in polynomial time such that