Fixing the Integrated Diffie-Hellman-DSA Key Exchange Protocol

Raphael C.-W. Phan, *Member, IEEE*

***Abstract*— Recently, three key exchange protocols were proposed that integrated the Diffie-Hellman key exchange into the Digital Signature Algorithm (DSA). It was claimed that the protocols provide known-key security and unknown key-share resilience, while the most advanced variant also provides key-replay resilience. However, we show in this Letter that the protocols do not provide forward secrecy and key freshness which are two of the standard security attributes that key exchange protocols should have. We also fix the protocols such that they provide these security attributes.**

***Index Terms*— Communication protocols, network security, Diffie-Hellman, Digital Signature Algorithm (DSA).**

I. Introduction

In 1993 [1], Arazi integrated the Diffie-Hellman (DH) key exchange protocol [2] with the digital signature algorithm (DSA) [3] to achieve mutual authentication of the exchanged keys. In 1994, Nyberg and Rueppel [4] showed that Arazi's proposal did not provide known-key security [5]. In 2004, Harn *et al.* [6] improved on Arazi's concept and presented 3 similar protocols that were shown to provide known-key security [5] and unknown key-share resilience [5], and the most advanced variant also provides key-replay resilience [6]. These three security attributes are standard requirements for any key exchange protocol, but they are not exhaustive.

In fact, as we will show in this Letter, Harn *et al.*'s protocols fail to provide two other standard security attributes, namely forward secrecy [5] and key freshness [5].

II. Harn *et al.*'s Key Exchange Protocols

Let p and q be two primes as in the normal DH case, and g be an integer where q is a divisor of $(p-1)$, while $g = h^{(p-1)/q} \bmod p$ and $g > 1$ for some random integer h with $1 < h < p-1$.

The private key of the user, U is a random value $x_U (0 < x_U < q)$ while $y_U = g^{x_U} \bmod p$ is the corresponding public key. $H(m)$ denotes a secure hash function on message m. $\{p, q, g, y_U\}$ are public values. The signature of m is the pair (r, s) computed as $r = ((g^k \bmod p) \bmod q)$ and $s = (k^{-1}(H(m) + x_U r)) \bmod q$, where k^{-1} denotes the multiplicative inverse of $k \bmod q$.

Harn *et al.* [6] proposed three variants of key exchange protocols that are very similar to each other. In this Letter, we will concentrate only on the most advanced version (the three-round protocol) and how our attacks apply to it since any successful attack on this version means the other two simpler ones are also susceptible.

For compactness of description, this protocol is illustrated in Fig. 1. v and w respectively denote short-term (ephemeral) secret values chosen by the two parties A and B respectively for that session.

III. Limitations of the Protocols

A. No Forward Secrecy

Forward secrecy [5] means that if a long-term private key is exposed, then the secrecy of previous established session keys should be maintained.

The session key for direction from A to B is computed by A as:

$$K_{AB} = y_B^v \bmod p, \tag{1}$$

while it is computed by B as:

$$K_{AB} = m_A^{x_B} \bmod p. \tag{2}$$

Therefore, when the long-term private key, x_B of B is compromised, an attacker can easily compute any previously established session key, K_{AB} by (2).

Similarly, the session key for direction from B to A is computed by A as:

$$K_{BA} = m_B^{x_A} \bmod p, \tag{3}$$

and computed by B as:

$$K_{BA} = y_A^w \bmod p, \tag{4}$$

Hence when the long-term private key, x_A of A is compromised, an attacker can easily compute K_{BA} by (3).

B. No Key Freshness

Key freshness [5] means that neither party can predetermine the shared secret key being established.

Again, all three versions of Harn *et al.*'s protocols do not provide any key freshness, meaning that both A and B can predetermine the shared secret key being established. This is true as follows.

A computes K_{AB} via (1), which depends on B's public key, y_B known by A all the time, and a random secret value, v chosen by A.

Therefore, A could have decided that K_{AB} must be equal to a predetermined value, namely $K_{AB} = y_B^v \bmod p$ where v was chosen at that point of time in the past. At any later time, whenever A wishes for K_{AB} to be that predetermined value, he simply uses that previously chosen v in forming the

Manuscript received November 22, 2004. The associate editor coordinating the review of this letter and approving it for publication was Chuan-Kun Wu.

Raphael C.-W. Phan is with the Information Security Research (iSECURES) Lab, Swinburne University of Technology (Sarawak Campus), Kuching, Malaysia.

Digital Object Identifier 10.1109/LCOMM.2005.06026.

Step	User A	User B
1.	Select random integer, v. $m_A = g^v \bmod p$.	
	$\longrightarrow m_A$	
2.		Select random integer, w. $K_{BA} = y_A^w \bmod p$. $K_{AB} = m_A^{x_B} \bmod p$. $m_B = g^w \bmod p$. $r_B = m_B \bmod q$. $s_B = (w^{-1} \times (H(m_B \| K_{BA} \| K_{AB}) + x_B r_B)) \bmod q$.
	$\longleftarrow m_B, s_B$	
3.	$K_{AB} = y_B^v \bmod p$. $K_{BA} = m_B^{x_A} \bmod p$. $r_B = m_B \bmod q$. Verify DSA signature (r_B, s_B) of message m_B. $r_A = m_A \bmod q$. $s_A = (v^{-1} \times (H(m_A \| K_{AB} \| K_{BA}) + x_A r_A)) \bmod q$.	
	$\longrightarrow s_A$	
4.		$r_A = m_A \bmod q$. Verify DSA signature (r_A, s_A) of message m_A.

Fig. 1. Three-round protocol

Step	User A	User B
1.	Select random integer, v. $m_A = g^v \bmod p$. $n_A = y_A^v \bmod p$.	
	$\longrightarrow m_A, n_A$	
2.		Select random integer, w. $K_{BA} = n_A^w \bmod p = g^{x_A vw} \bmod p$. $K_{AB} = m_A^{x_B w} \bmod p = g^{x_B vw} \bmod p$. $m_B = g^w \bmod p$. $n_B = y_B^w \bmod p$. $r_B = m_B \bmod q$. $s_B = (w^{-1} \times (H(m_B \| K_{BA} \| K_{AB}) + x_B r_B)) \bmod q$.
	$\longleftarrow m_B, n_B, s_B$	
3.	$K_{AB} = n_B^v \bmod p = g^{x_B vw} \bmod p$. $K_{BA} = m_B^{x_A v} \bmod p = g^{x_A vw} \bmod p$. $r_B = m_B \bmod q$. Verify DSA signature (r_B, s_B) of message m_B. $r_A = m_A \bmod q$. $s_A = (v^{-1} \times (H(m_A \| K_{AB} \| K_{BA}) + x_A r_A)) \bmod q$.	
	$\longrightarrow s_A$	
4.		$r_A = m_A \bmod q$. Verify DSA signature (r_A, s_A) of message m_A.

Fig. 2. Fixed three-round protocol

$m_A = g^v \bmod p$ to B. This will cause K_{AB} to be equal to the predetermined value, y_B^v.

Similarly, B computes K_{BA} via (4) and so he could choose this to be equal to any predetermined value, $K_{BA} = y_A^w \bmod p$ by using a previously chosen value of w in forming his message m_B to A.

A and B therefore can predetermine at a certain time in the past, what a future session key, K_{AB} and K_{BA} respectively would be equal to.

As an aside, note that if A or B do this, then they are putting the confidentiality of their long-term private key at risk. This is because doing so will necessitate two different DSA signatures to be generated using the same random value − this is known to have an associated risk of key compromise, if any other party spots this.

IV. Fixing the Protocols

We can fix the protocols so that both forward secrecy and key freshness can be guaranteed, while preserving the basic essence of the original protocols. This is shown in Fig. 2.

The main idea is to ensure the computations of the two session keys, K_{AB} and K_{BA} depend on the ephemeral secrets, v and w chosen by both parties A and B.

This provides forward secrecy because even if the long-term private key of any party is exposed, previous session keys cannot be computed since the ephemeral secrets, v and w for that session are unknown.

This fix also provides key freshness because every session key is a function of ephemeral secrets chosen by both parties, so neither party can predetermine a session key's value since he would not know what the other party's ephemeral secret is going to be.

V. Conclusion

We have shown that the most advanced version of Harn *et al.*'s three key exchange protocols fails to provide two standard security criteria that are required of any key exchange protocol, namely forward secrecy and key freshness, and it is easy to extend the result to all three versions proposed by them [6]. We have also shown how to fix the protocols to provide forward secrecy and key freshness.

Acknowledgement

We would like to thank the anonymous referees for their helpful suggestions and comments which greatly improved this Letter. We also thank God for His many blessings (Ps. 111).

References

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

[2] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 644-654, Nov. 1976.

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

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

[5] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, FL: CRC Press, 1997.

[6] 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.