International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463
Vol. 3 Issue 8, August-2014, pp: (113-116), Impact Factor: 1.252, Available online at: www.erpublications.com

Authenticated Diffie-Hellman Key Exchange with Forward Secrecy

Y. Venkatramana Reddy[1], Dr. G. V. Satyanarayana[2], Dr. M. Venkateswara Rao[3]

[1]Member ACM, IACR, GITAM University Visakhapatnam, India
[2]Professor & HOD, Dept. of IT, Raghu Institute of Technology and Science, Vizag, India
[3]Associate Professor, GITAM University, Visakhapatnam, India

Abstract: Forward secrecy is an important security property in key agreement protocol. Based on Harn's protocol, in this paper a new authenticated Diffie-Hellman key agreement protocol with half forward secrecy is proposed. This protocol is also based on a single cryptographic assumption, and is user authentication and shared key authentication. More importantly, our protocol provides forward secrecy with respect to one of the parties. For this reason, besides the advantages of Harn's rotocol, in practice, our protocol can reduce the damages resulted from the disclosure of the user's secret key and it is very beneficial to today's communication with portable devices.

Keywords: cryptographic protocol; authenticated Diffie-Hellman key agreement protocol; forward secrecy.

1 Introduction

Key agreement is an essential primitive in secure communication for establishing session keys. In 2005, after many improvement of authenticated Diffie-Hellman key agreement protocol[1-5], Harn et al proposed an authenticated Diffie-Hellman key agreement protocol using a single cryptographic assumption[6], called HHM protocol. This protocol has many advantages, such as being based on a single cryptographic assumption and being user authentication as well as shared-key authentication. Typically, it avoids the attacks resulted from hash functions[7].Yet, in spite of its attractiveness and success, we find that HHM protocol does not provide forward secrecy which is an important security property in today's communication.

A key agreement protocol is said to be forward secure if the compromise of one or more parties' long-term keys does not compromise past session keys[8]. In other words, if a key agreement protocol does not support forward secrecy, the compromise of one or more parties' long-term keys will result in the compromise of past session keys. There are two distinct scenarios of forward secrecy: half forward secrecy (if a single party's private key is compromised) and full forward secrecy (the private keys of both parties are compromised). In practice nowadays, as portable devices are spreading widely, there are more occasions for exposing a secret key to adversaries. To mitigate the damage caused by the exposure of secret keys stored on such devices, cryptographic protocols are basically required to meet the notion of forward security.

To achieve forward secrecy, based on HHM protocol, the authors of this paper propose a new authenticated Diffie-Hellman key agreement protocol, and prove that this protocol supports half forward secrecy. Thanks to this contribution, our new protocol can reduce the damage caused by the exposure of secret keys stored on portable devices. So the protocol our proposed is very suitable for today's secure communication.

2 HHM Protocol and its Security Analysis

2.1 HHM Protocol

HHM protocol can be divided into two stages: The initiation stage and the key agreement stage. The two stages are described as follows:

During the initiation stage, the system chooses and publishes a large prime number p and a primitive element g in GF(P) . As the Diffie-Hellman protocol in Ref.[9] describes, each user in the system selects a long-term secret key **x** and computes a corresponding long-term public key $y = g^x \bmod p$. Let Alice and Bob be two users in the system. The long-term secret key and the long-term public key for Alice are X_A and Y_A. Similarly, those for Bob are X_B and Y_B, respectively. In addition, cert(YA) is the certificate of YA and cert(Y_B) is that of Y_B . Subsequently, Alice and Bob may use the following key agreement protocol to establish their common session keys.

Step 1: Alice chooses a random integer a called the short-term secret key, and computes the corresponding short-term public key $r_A = g^a \bmod p$, then sends the messages r_A to Bob.

Step 2: Similarly, Bob chooses b, computes:
$r_B = g^b \bmod p$
$K_{AB} = (r_A)^{XB} \bmod p$
$S_B = K_{AB}^{-1}(X_B - br_B) \bmod p$
then sends (r_B, S_B) to Alice.

Step 3: Upon receiving (r_B, S_B) , Alice verifies Y_B by checking cert(Y_B), then computes:
$K_{AB}^1 = (Y_B)^a \bmod p$
and verifies S_B and K_{AB}^1 by checking whether the equation $Y_B = (r_B)^{rB} (g)^{SBKAB1} \bmod p$ holds. If it holds, Alice computes:
$K_{BA} = (r_B)^{XA} \bmod p$.
$S_A = K_{BA}^{-1}(X_A - ar_A) \bmod p$
and sends S_A to Bob.

Step 4: Bob verifies Y_A by checking cert(Y_A) , computes:
$K_{BA}^1 = (Y_A)^b \bmod p$
and verifies S_A and K_{BA}^1 by checking wether the $Y_A = (r_A)^{rA}(g)^{SAKBA1} \bmod p$ holds.
Thus, Alice and Bob obtain two common secret keys as the following:
$K_{BA} = (r_A)^{XB} \bmod p = (Y_B)^a \bmod p = K_{AB}^1$
$K_{AB} = (r_B)^{XA} \bmod p = (Y_A)^b \bmod p = K_{BA}^1$

2.2 Security Analysis of HHM Protocol

Two common attributes desired for key agreement protocols pointed out by the Ref.[10] are:
Authentication: An agreed-upon secret key should be known (or knowable) only by identified parties;
Forward secrecy: An agreed-upon secret key should remain secret, even if one or more parties' long-term key material is compromised.

2.2.1 Authentication

In modern communication protocols, there are two types of authentication, namely, user authentication and shared-key authentication. User authentication is to authenticate a communicating user in real time. In a key agreement protocol without user authentication, an attacker can misrepresent the identity of an innocent party, leading to the attacks such as replay attack and unknown key-share attack. Shared-key authentication ensures that a shared key is known only to the legitimate users. A key agreement protocol without either user authentication or shared-key authentication is not secure, leading to many kinds of attacks. Therefore a secure key agreement protocol needs both user authentication and shared-key authentication.
HHM protocol provides both user authentication and shared-key authentication as described in Ref.[6]. So, it prevents known key attacks and impersonation attacks etc.

2.2.2Forward secrecy

For authenticated key agreement protocol, forward secrecy is about the protection of previously established session keys after the participants' long-term private keys are compromised.
In HHM protocol, Alice and Bob compute the session key as:
$K_{AB} = (r_A)^{XB} \bmod p$
$K_{BA} = (r_B)^{XA} \bmod p$
Obviously, anyone who obtains X_B can get K_{AB} , and anybody who obtains X_A can get K_{BA} . So, if one of the

Parties' long-term private keys, X_A or X_B, is compromised, the session key will no longer remain secret. In other words, HMM protocol does not provide forward secrecy with respect to any participant in communication.

3 A New Authenticated Diffie-Hellman Key Agreement Protocol

In practice, forward secrecy becomes more and more important in today's communication, and a protocol without forward secrecy will be very vulnerable. So, based on the HHM protocol, we propose a new authenticated Diffie-Hellman key agreement protocol using a single cryptographic assumption, which provides forward secrecy with respect to one party's long-term private key.
The initiation stage of the new protocol is the same as HHM protocol. The following is its key agreement stage.

Step 1: Alice verifies YB by checking Cert(YB) and chooses a random integer a, and computes:

$K_A = (Y_B)^a \bmod p$
$S_A = \backslash a/(K_A + X_A) \bmod q$
$R_A = g^{KA} \bmod p$
Alice sends the message (R_A, S_A) to Bob

Step 2: On receipt of (R_A, S_A), Bob Verifies Y_A by Cert(Y_A), then computes
$K_A = (R_A Y_A)^{XBSA}$, and checks whether the equation $R_A = g^{KA} \bmod p$ holds. If it holds, Bob chooses a random integer b, and computes:
$K_B = (Y_A)^b \bmod p$
$S_B = b/(K_B + X_B) \bmod q$
$R_B = g^{KB} \bmod p$
then Bob obtains $K = K_B K_A$, and sends (R_B, S_B) to Alice.

Step 3: Upon receiving (R_B, S_B), Alice computes
$K_B = (R_B Y_B)^{XASB}$ and checks whether $R_B = g^{KB} \bmod p$ holds. If it holds, Alice obtains $K = K_A K_B$.
Thus, Alice and Bob can obtain the common session key K.
The correctness of our protocol can be proved as follows.

$K = K_B K_A = K_B (Y_B)^a$
$= K_B g^{aXB} \bmod p$
$= K_B g^{(KA + XA)SAXB} \bmod p$
$= K_B g^{(KASAXB)} g^{(XASAXB)}$
$= K_B (R_A Y_A)^{XBSA}$

$K = K_A K_B$
$= K_A (Y_A)^b \bmod p$
$= K_A g^{bXA} \bmod p$
$= K_A g^{(KB + XB)SBXA} \bmod p$
$= K_A g^{KBSBXA} g^{XASAXB}$
$= K_A (R_B Y_B)^{XASB}$

4 Security Properties of the New Protocol

Firstly, the security of the new protocol is entirely based on DL assumption. This is more desired by recent researchers. As such it avoids more threats resulted from other cryptographic assumptions.

Secondly, our protocol achieves shared-key authentication. After receiving (R_A, S_A) from Alice, Bob calculates K by $K = K_B (R_A Y_A)^{XBSA}$; similarly, after receiving(R_B, S_B) from Bob, Alice calculates K by $K = K_A (R_B Y_B)^{XASB}$. Because X_B is private to Bob and X_A is private to Alice, K is only known to users Alice and Bob. Thus our protocol provides shared-key authentication.

Thirdly, our protocol provides user authentication. Our protocol is based on signature authentication and it convinces Alice that (R_B, S_B) is really generated by Bob; moreover it convinces that K is not a replayed key because of K_B being a computed number based on a random number b. Thus, if Alice successfully verifies the signature, Alice can be sure that (R_B, S_B) is not a replayed

message, and hence K is not a replayed key. A similar analysis of user authentication can be applied from user Bob's point of view. Thus, our protocol provides mutual user authentication between the two communication parties.

Due to both user authentication and shared-key authentication, our proposed protocol prevents known key attacks, replay attacks, and unknown key-share attacks.

Besides these advantages, the most important is that our protocol provides forward secrecy with respect to one of the participants, i.e. half forward secrecy. The proof is as follows.

In our protocol, the session key between Alice and Bob is

$K = K_A (R_B Y_B)^{XASB} = K_B (R_A Y_A)^{XBSA}$
$= (R_B Y_B)^{XASB} (R_A Y_A)^{XBSA}$

According to the above formulae, to compute the session key, one must simultaneously have X_A and X_B, or X_A and K_A or X_B and K_B. So, if Alice's long-term private key X_A is compromised, nobody (except Bob) can get the session key even if someone obtains X_A because he or she doesn't know K_A and K_B. So the session key remains secret. The analogous analysis can be done for the case that Bob's long-term private key X_B is compromised. These mean that our protocol provides half forward secrecy. In other words, in our protocol, even if long-term private key of one participant is compromised, the secrecy of the previous session key established by honest entities is not affected.
In particular, our protocol's computation cost and communicational overhead is obviously the same with HHM.

Conclusions

Based on the recent authenticated Diffie–Hellman key agreement protocol using a single cryptographic assumption, this paper proposes a new authenticated Diffie–Hellman key agreement protocol with half forward secrecy. The protocol is entirely based on a single cryptographic assumption and provides user authentication and shared-key authentication. Typically, it provides forward secrecy with respect to one of the parties while its efficiency is the same

with HHM protocol. When the new protocol is used to establish session keys in communication, the secrecy of the session keys from earlier runs will not be compromised even if one party's long-term secret key is disclosed. So the new protocol is very suitable for today's application, as a number of cryptographic computations are performed on portable devices that are more vulnerable to attacks.

References

[1]. Menezes A J, Qu M, Vanstone S A. Some Key Agreement Protocols Providing Implicit Authentication[C] 2nd Workshop Selected Areas in Cryptography. Berlin: Springer-Verlag, 1995: 89-98.
[2]. IEEE 2000, IEEE Std. 1363-2000: Standard Specifications for Public Key Cryptography[S/OL]. [2007-11-20].
[3]. IEEE P1363 Working Group, 2001. IEEE P1363a D10 (Draft Version 10): Standard Specifications for Public Key Cryptography:Additional Techniques[S/OL]. [2007-11-20]
[4]. Hugo K. HMQV: A High-Performance Secure Diffie-Hellman Protocol (Extended Abstract) [J]. International Association for Cryptologic Research , 2005, 3621: 546-566.
[5]. Harn L, Lin H Y. Authenticated Key Agreement Protocol without Using One-Way Function [J]. Electronics Letters, 2001, 37 (10): 629-630.
[6]. Harn L, Hsin W J, Mehta M. Authenticated Diffie-Hellman Key Agreement Protocol Using a Single Cryptographic Assumption [J]. IEE Proc Commun, 2005, 152(4): 405-410.
[7]. Dobbertin H. The Status of MD5 after a Recent Attack [J]. CryptoBytes, 1996, 2(2): 1-6.
[8]. Menezes A J, van Oorschot C, Vanstone S A. Handbook of Applied Cryptography[M]. Boca Raton: CRC Press, 1997.
[9]. Diffie W, Oorschot P C V, Wirner M J. Authentication and Authenticated Key Exchanges [J]. Codes Cryptography, 1992, 2: 107-125.
[10]. Burton S, Kaliski J R. An Unknown Key-Share Attack on the MQV Key Agreement Protocol [J]. ACM Transactions on Information and System Security, 2001, 4(3): 275-288.