文章编号:1004-1478(2011)03-0053-05

一种改进的抗攻击密钥协商协议研究

张珂

(陕西师范大学 网络信息中心,陕西 西安 710062)

摘要:针对Diffie-Hellman密钥交换协议缺乏对通信双方身份的认证而导致该协议易受到中间人攻击和重放攻击的问题,提出了一种改进的Diffie-Hellman密钥交换协议——AADH协议.性能分析表明,AADH协议由于引入了密钥认证及时间戳和随机的通信标号等机制、继承了Diffie-Hellman协议的安全性能,具有较高的抗中间人攻击和重放攻击的能力.

关键词:Diffie-Hellman密钥交换协议;中间人攻击;重放攻击;签名;认证

中图分类号:TP393.08 **文献标志码**:A

Study on a new anti-attack key agreement protocol

ZHANG Ke

(*Network Infor. Center, Shaanxi Normal Univ., Xi'an* 710062, *China*)

Abstract: Since the Diffie-Hellman key exchange protocol lacks the authentication for identity, it might be suffered from the man-in-the-middle and replay attack. In order to resolve this bug, a modified Diffie-Hellman key exchange protocol—AADH protocol was put forward. The AADH protocol has introduced some new technologies, such as key authentication, timestamps and random numbers which is the marker of communication. The analysis of the performance showed that due to the safe performance of Diffie-Hellman protocol, AADH protocol has high ability in fighting back the man-in-the-middle and replay attacks.

Key words: Diffie-Hellman key exchange protocol; man-in-the-middle attack; replay attack; autograph; authentication

0 引言

2010年7月由中国互联网络信息中心公布的《中国互联网络发展状况统计报告》中指出:截至2010年6月,中国的网民数已增至4.2亿人,规模跃居世界第一位.由此可见,随着计算机和互联网技术的普及,网络通信已经渗透到社会的各个方面.信息安全不仅仅是政府和军事部门高度关注的问题,对企事业单位也同样重要.信息安全与隐私的保护程度已逐渐成为衡量互联网技术的关键指标.相关民意调查[1]表明,用户使用互联网时最大的担心就是自己的隐私被发现.保证信息在发送方与接收方之间传送时不被窃密者窃取并破译,是网络技术人员以及密码学家们所应负的责任.要想使信息可靠传输,发信者必须将所发的数据(即明文)通过加密系统变成密文,收信者收到密文后再用相

收稿日期:2010-12-09

作者简介:张珂(1982—),男,陕西省渭关县人,陕西师范大学助理工程师,主要研究方向为网络安全.

应的解密系统对密文解密从而恢复成明文,密码学在网络安全方面发挥着越来越重要的作用[2-3].

信息安全以密码学技术为基础,但密码学中最关键的问题是密钥的分发和管理. Diffie-Hellman 密钥交换协议(以下简称 DH 协议)较好地解决了密钥分发的问题,但是由于在密钥协商时并没有对协议主体进行身份认证,因此 DH 协议易受到中间人攻击[4]和重放攻击[5],所以 DH 协议在实际应用时,必须对其进行改进. 文献[6]在 Java 平台下实现 DH 协议,将 DH 协议的运行过程形象地展示给用户,便于用户了解 DH 协议的运行过程及运行原理.

针对 DH 协议的安全性缺陷,本文基于密钥认证机制提出改进的 Diffie-Hellman 密钥协商协议——AADH (autograph authentication Diffie-Hellman)协议,并对 AADH 协议性能进行详细分析.

1 Diffie-Hellman 密钥交换协议

Diffie 和 Hellman 在一篇具有独创意义的论文中首次提出公钥算法,给出了公钥密码学的定义,该算法通常称为 Diffie-Hellman 密钥交换协议. 该算法的目的是使 2 个用户能安全地交换密钥,以便在后续的通信中用该密钥对消息进行加密.

1.1 DH 协议

DH 协议的密钥交换算法中 q 是大素数,a 为 q 的本原根,在 Alice 与 Bob 通信前都已知公开的 q 和 a. 假定用户 Alice 和 Bob 希望交换密钥,用户 Alice 选择一个保密的随机整数 $X_A < q$,并计算 $Y_A = a^{X_A} \bmod q$ 发送给 Bob,类似地,Bob 选择一个保密的随机整数 $X_B < q$,并计算 $Y_B = a^{X_B} \bmod q$ 发送给 Alice. Alice 和 Bob 保持其 X 是私有的,但对另一方而言,Y 是公开可访问的,用户 Alice 计算 $K = (Y_B)^{X_A} \bmod q$ 并将其作为密钥,用户 Bob 计算 $K = (Y_A)^{X_B} \bmod q$ 并将其作为密钥. 因为

$$K = (Y_B)^{X_A} \bmod q = (a^{X_B} \bmod q)^{X_A} \bmod q =$$
$$(a^{X_B})^{X_A} \bmod q = a^{X_B X_A} \bmod q = (a^{X_A})^{X_B} \bmod q =$$
$$(a^{X_A} \bmod q)^{X_B} \bmod q = (Y_A)^{X_B} \bmod q$$

所以 Alice 和 Bob 持有相同的密钥.

1.2 DH 协议的不足

由于 DH 协议在通信过程中并未对参与通信双方的身份进行认证,所以 DH 协议不能抵抗中间人攻击,并且易受到重放攻击.

图 1 所示为 DH 协议中间人攻击模型,假定 Alice与 Bob 希望交换密钥,而 Darth 是攻击者.

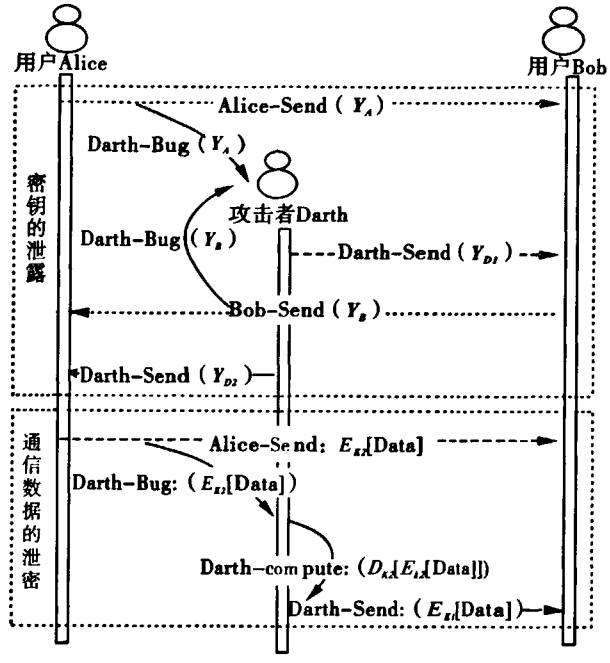

图 1 Diffie-Hellman 协议中间人攻击

攻击过程如下.

1.2.1 密钥的泄露

Alice →Bob: Alice-Send (Y_A);

Darth: Darth-Bug(Y_A);

Darth: Darth-compute(K_2);

Darth →Bob: Darth-Send(Y_{D1});

Bob: Bob-compute(K_1);

Bob → Alice: Bob-Send(Y_B);

Darth: Darth-Bug(Y_B);

Darth: Darth-compute(K_1);

Darth → Alice: Darth-Send(Y_{D2});

Alice: Alice-compute(K_2).

1)为了进行攻击,Darth 首先生成 2 个随机数 X_{D1} 和 X_{D2},然后计算相应的公钥 Y_{D1} 和 Y_{D2}.

2) Alice 将 Y_A 传递给 Bob.

3) Darth 截获了 Y_A,用 Y_{D1} 替代 Y_A,将 Y_{D1} 传递给 Bob. 同时 Darth 计算 $K_2 = (Y_A)^{X_{D2}} \bmod q$.

4) Bob 收到 Y_{D1},计算 $K_1 = (Y_{D1})^{X_B} \bmod q$.

5) Bob 将 Y_B 传递给 Alice.

6) Darth 截获 Y_B,用 Y_{D2} 替代 Y_B,将 Y_{D2} 传递给 Alice. 同时 Darth 计算 $K_1 = (Y_B)^{X_{D1}} \bmod q$.

7) Alice 收到 Y_{D2},计算 $K_2 = (Y_{D2})^{X_A} \bmod q$.

此时,Alice 和 Bob 想他们已共享了密钥,但实

际上,Bob 和 Darth 共享密钥 K_1,而 Alice 与 Darth 共享密钥 K_2. 图 1 中虚线表示由于遭到窃听并未成功的通信过程.

1.2.2 通信内容的泄密 Darth 成功与 Alice 和 Bob 建立通信后,Alice 与 Bob 之间的通信将会泄密:

Alice →Bob:Alice-Send(E_{K2}[Data]);

Darth:Darth-Bug(E_{K2}[Data]);

Darth:Darth-compute(D_{K2}[E_{K2}[Data]]);

Darth →Bob:Darth-Send(E_{K1}[Data]);

或 Darth →Bob:Darth-Send(E_{K1}[Data']).

1) Alice 发送一份加密消息 M:E_{K2}[Data].

2) Darth 截获该加密消息,解密,恢复出数据信息 Data.

3) Darth 将 E_{K1}[Data]或 E_{K1}[Data']发给 Bob,其中信息 Data'是任意的消息. 第 1 种情况,Darth 只是简单地窃听 Alice 与 Bob 间的通信,而对通信内容不做修改. 第 2 种情况,Darth 修改了 Alice 发往 Bob 的信息.

2 AADH 协议

针对目前 DH 协议存在的上述不足,本文将设计基于签名认证机制的改进的 DH 协议——AADH 协议,如图 2 所示,其中 Alice,Bob 为密钥协商用户,KAC 为密钥认证中心,向密钥协商用户颁发身份证书. 该协议建立了一个完整的密钥协商及密钥认证过程.

AADH 协议会话过程如下所示.

2.1 密钥的协商

Alice:Alice-Random (X_A);

Alice:Alice-Compute(Y_A);

Alice →Bob:Alice-Send($Y_A \| TS \| N_a$);

Bob:Bob-Random (X_B);

Bob:Bob-Compute(Y_B);

Bob → Alice:Bob-Send($Y_B \| TS \| N_b$);

Alice:Alice-Compute(K);

Bob:Bob-Compute(K).

Random()为随机数产生函数,Compute()为公钥、密钥计算函数. TS 为时间戳,N_a,N_b分别为 Alice 和 Bob 产生的随机交互号.

2.2 密钥可信性验证

如图 2 所描述,Alice 与 Bob 利用 Compute()函数计算出密钥后,为防止密钥协商过程遭到中间人攻击,分别向密钥认证中心 KAC 发出密钥验证

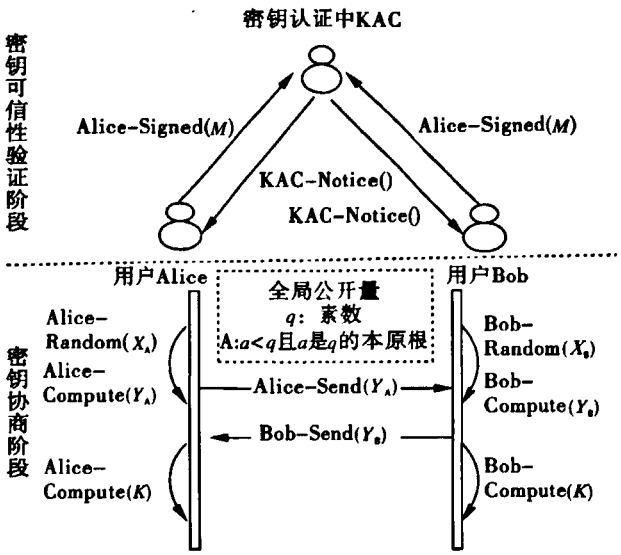

图 2　AADH 协议模型

请求. 首先 KAC 通过颁发的身份证书验证 Alice 和 Bob 身份的合法性,拒绝响应非法用户的密钥认证请求. 由于 Alice 和 Bob 应具有相同的密钥,KAC 在接收到双方的密钥认证请求后,对双方发送的密钥进行合法性验证:若 Alice 与 Bob 所持有的密钥一致,KAC 则向双方返回合法性验证通过的消息;否则,KAC 发出警告,告知 Alice 与 Bob 密钥协商过程可能遭受中间人攻击,通知双方进行新的密钥协商. 同时 Alice 和 Bob 在向 KAC 发送密钥认证请求时,为免遭窃听等攻击,所发的消息带有签名信息,KAC 可以通过签名来验证请求消息的合法性.

1) Alice 的密钥认证请求.

Alice →*KAC*:*Alice-Signed*(*Message* || *R* || *S* || *TS* || *Life-Time*)

Message = [*Source-Address* || *Purpose-Address* || *K*]

其中,*Source-Address* 为 Alice 的地址,*Purpose-Address* 为与 Alice 协商密钥的用户地址(即 Bob 的地址),K 为 Alice 的密钥.

公式①②的计算结果 R 和 S 组成签名信息,其中,TS 为时间戳,*Life-Time* 为密钥认证请求消息的生命周期.

$$R = (g^k \bmod a) \bmod q \qquad ①$$

$$S = (K^{-1}(H(Message) + X_A R)) \bmod q \qquad ②$$

其中,$H(Message)$为使用 SHA-1 求 *Message* 的散列码;q,a 和 X_A为密钥交换过程所产生的参数,q 为素数,a 为 q 的本原根;X_A,k 为随机数;g 和 h 满足 $h^{(a-1)/q} \bmod a$ 且 $g>1$. Bob 的请求认证过程类似,

本文不再赘述.

2)KAC验证请求信息的合法性.

KAC首先通过公式③—⑥的计算验证请求信息是否被破坏.

$$W=(S')^{-1}\bmod q \quad ③$$

$$U_1=[H(Message')W]\bmod q \quad ④$$

$$U_2=(R')W\bmod q \quad ⑤$$

$$V=[(g^{U_1}y^{U_2})\bmod a]\bmod q \quad ⑥$$

S',$Message'$和R'表示接收到的S,$Message$和R.通过检验V与R'是否相等来判断请求信息是否遭到破坏.

若一方的请求先到,则将其在KAC的缓存区中等待一段时间:该段时间内,对方的请求到达,KAC进行验证;否则请求超时,本次验证失败,即公式⑦成立时请求验证超时,KAC通知双方进行新的密钥协商.

$$|当前时钟-Time|>\Delta T_1+\Delta T_2+Life\text{-}Time \quad ⑦$$

其中,$Time$为请求到达时间,ΔT_1为时钟误差,ΔT_2为网络时延,$Life\text{-}Time$为请求的生命周期.

3)密钥真实性验证.

当KAC接收到密钥交换双方完整的请求信息后,对双方的密钥进行一致性验证,并向双方通知相应的验证结果.KAC的验证算法为

```
If ( |Now-Time-Time| > ΔT1 + ΔT2 + Life-Time)
  KAC →Alice:KAC-Notice(False || INF-Type-1);
Else
 If ( (Alice →V = = Alice →R')&& (Bob →V = = Bob
   →R') )
   If (Alice →K = = Bob →K)
     KAC →Alice:KAC-Notice(True);
   Else
     KAC →Alice:KAC-Notice(False || INF-Type-2);
   End If
 Else
   KAC →Alice:KAC-Notice(False || INF-Type-3);
 End If
End If
```

其中,$INF\text{-}Type$为认证失败返回消息类型.

3 AADH协议的性能分析

3.1 中间人攻击分析

由图1所知,AADH协议若被中间人攻击成功,即双方协商的密钥通过KAC的验证,则应有Darth与Alice间的密钥K_2和Darth与Bob间的密钥K_1相等,即$K_1=K_2$,要使$K_1=K_2$,则Darth产生的随机数X_{D1}和X_{D2}分别和Alice与Bob产生的随机数X_A和X_B相等,即$X_{D1}=X_A$,$X_{D2}=X_B$.

DH协议中q为素数,随机整数X_{D1},X_A,X_{D2}和X_B分别满足$X_{D1}<q$,$X_A<q$,$X_{D2}<q$,$X_B<q$,假设用N表示小于q的所有整数的个数,用$P_{(X=Y)}$表示随机整数X等于Y的概率,即有$P_{(X=Y)}=1/N$.则有$X_{D1}=X_A$,$X_{D2}=X_B$的概率分别为:$P_{X_A}=X_{D1}=1/N$和$P_{X_B}=X_{D2}=1/N$,Darth攻击成功的概率为$P_{(X_A=X_{D1})\text{and}(X_B=X_{D2})}=1/N^2$.

当素数q的值越大时,N值越大,即当$q\to\infty$时$N\to\infty$,则有$\lim\limits_{q\to\infty}P_{(X_A=X_{D1})\text{and}(X_B=X_{D2})}=\lim\limits_{N\to\infty}\frac{1}{N^2}=0$;当$q$为大素数时,Darth攻击成功的概率为0,如图3所示.所以说AADH协议能够抵抗中间人攻击.

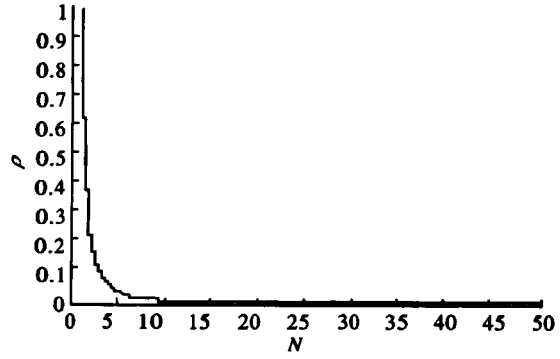

图3 AADH协议中间人攻击成功的概率仿真图

3.2 功能分析

AADH协议保证了DH协议原有的功能,没有因为方案的改动发生变化.AADH协议弥补了DH协议易受到中间人攻击和重放攻击的缺陷,AADH协议通过用户向密钥认证中心KAC发送认证请求信息的方式,达到抵制中间人攻击的目的.改进后的DDAH协议在完成密钥交换的同时可确保该过程的可信性及密钥的可靠性.

3.3 安全性分析

DH协议本身具有安全性能高的优点,在DH协议中X_A和X_B是保密的,敌对方可以利用的参数只有q,a,Y_A和Y_B,因而敌对方被迫取离散对数来确定密钥.如要获知用户Bob的秘密密钥,敌对方必须先计算:$X_B=d\log_{a,q}(Y_B)$,然后再使用用户Bob采用的方法计算其秘密密钥K.DH密钥交换算法的安全性依赖于这样一个事实:虽然计算以1个素数为模的指数相对容易,但计算离散对数却很困难.对于

大的素数,计算出离散对数几乎是不可能的.

在AADH协议中引入密钥认证中心KAC机制来对用户协商的密钥进行合法性的验证,确保密钥协商过程的可信性,密钥协商阶段引入时间戳和随机的通信标号能够防止重放攻击.因此AADH协议能够弥补DH协议所存在的易受重放攻击和中间人攻击的缺陷.

首先,AADH协议解决了DH协议易受到中间人攻击的缺陷.DH协议有这个缺陷是因为该协议只能保证协商密钥的双方协商得到共享密钥,却不能保证协商密钥的双方都具有合法身份,攻击者就可以冒充另一方分别与原双方进行密钥协商,从而实现中间人攻击.AADH协议引入密钥认证中心KAC机制对用户协商密钥的合法性及其真实性进行验证,这样就克服了DH协议易受到中间人攻击这一缺陷.

其次,AADH协议解决了DH协议易受重放攻击的缺陷,在密钥协商阶段,AADH协议中通信过程引入时间戳和随机的通信标号机制,通过时间戳防止重放攻击.但鉴于用户间时间系统同步的难以实现性,AADH协议中又引入随机的通信标号,因为标号的随机性及对对方的不可预知性,更能抵抗重放攻击.

由此可见,AADH协议在继承DH协议优点的同时,能够解决DH协议易受到中间人攻击和重放攻击的问题.

3.4 效率分析

由于KAC只是对用户的密钥进行简单的一致性验证,并没有进行其他复杂操作,同时由于指数运算$g^k \bmod a$不依赖于被签名的消息,可以提前计算,所以AADH协议在改进DH协议的同时基本保留了DH协议原有的效率.

4 结语

密钥交换是实现网络安全通信的第一步,也是最重要的环节.本文提出了一种对DH协议的改进协议——AADH协议,该协议通过引入密钥认证、时间戳和随机的通信标号等技术来抵抗中间人攻击和重放攻击,可弥补DH协议的不足.下一步研究方向是基于可信计算理论研究KAC的安全性保护,以防止KAC由于遭受到攻击而报告虚假的验证信息.

参考文献:

[1] Claessens J, Diaz C, Goemans C. Revocable anonymous access to the Internet[J]. Int Research Electr Networking Appli and Policy, 2003(2):13.

[2] Atul Kahate. 密码学与网络安全[M].2版.金名译.北京:清华大学出版社,2009.

[3] Stallings W. 密码编码学与网络安全——原理与实践[M].4版.孟庆树,王丽娜,傅建明,等译.北京:电子工业出版社,2008.

[4] 方俊. 一种基于Diffie-Hellman密钥交换协议的OTP方案[J].计算机时代,2009(11):24.

[5] 张世勇. 网络安全原理与应用[M].北京:科学出版社,2003.

[6] 尹少平,董丹.Diffie-Hellman密钥交换协议设计与实现[J].电力学报,2006,24(1):9.