一种用于口令同步的通用混合密码传输协议

李华峰, 钱焕延

(南京理工大学 网络中心, 江苏 南京 210094)

[摘要] 结合基于对称密钥的认证协议和基于公开密钥的认证协议, 提出了一种用于口令同步的通用混合密码传输协议, 给出了具体的需求分析、算法选择和协议内容, 并在协议分析的基础上用 BAN 逻辑证明了它的有效性. 分析结果证明, 该协议能够达到预期目标.

[关键词] 网络安全, 认证协议, 口令同步, 混合密码传输协议

[中图分类号] TP393 12 [文献标识码] A [文章编号] 1672-1292(2008)04-0178-04

A General Hybrid Cryptograph Transfer Protocol Applied in Password Synchronization

Li Huafeng Qian Huanyan

(Network Center, Nanjing University of Science and Technology, Nanjing 210094, China)

Abstract This paper combines the authentication protocol based on symmetric key with the authentication protocol based on public key, provides a general hybrid cryptograph transfer protocol(HCTP) applied in password synchronization. It also reveals requirements analysis, algorithm selection, protocol content, and proves its validity through BAN logic on the basis of protocol analysis. The analysis result proves that the purpose of protocol can be achieved as expected.

Key words network security, authentication protocol, password synchronization, HCTP

口令同步既可用于多个 Web服务器之间的同步, 也可用于不同类型应用服务器之间的同步. 我们以不同类型应用服务器之间的口令同步为例给出安全需求模型, 如图 1所示. 其中 S_i 表示第 i 个应用服务器 ($i = 1, 2, \cdots$), U_i 表示第 i 个应用服务器处需要上传的数据库. 按照一定的时间间隔, 应用服务器将自己的口令数据库发送给代理服务器.

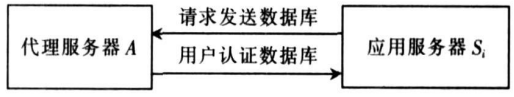

图 1 口令同步模型图

Fig.1 Password synchronization model

安全传送认证数据库要求: 信息可鉴别性 (双方能够进行有效的身份认证); 保密性 (数据传输过程中必须对数据加以保护); 数据完整性 (保证数据在传输过程中没有被篡改). 为进行实时性较强的安全通信, 我们选用基于 Diffie-Hellman的密钥交换算法[1] 来产生主密钥. 但 Diffie-Hellman协议不能有效防止中间人攻击, 且不能进行会话双方的身份认证, 所以在实现时采用借助于数字证书的站间协议[2] 来进行双向认证, 并选用 RSA 算法进行数字签名产生数字证书. 尽管用 Diffie-Hellman密钥交换算法产生会话主密钥比用 RSA 密钥交换算法脆弱, 但该协议算法简洁、运算速度快[3].

1 算法选择

1.1 选定算法参数

(1) 系统管理员 Root选定自己的 RSA密钥对 (e_R, d_R) 和公共模数 n_R, 其中 e_R, n_R 为公钥, d_R 为私钥, 为了保证 RSA 算法的安全性, n_R 应满足: $n_R > 2^{1024}$;

收稿日期: 2008-06-18

通讯联系人: 李华峰, 博士研究生, 研究方向: 网络安全及网络应用技术. E-mail: lhfeng@mail.njust.edu.cn

(2) Root选定一种单向杂凑函数 $h(\cdot)$, 为保证算法的安全性, 选用公开算法, 如 MD5或 SHA;

(3) 选定一个大素数 p 及 GF(p) 上的一个生成元 g;

(4) 选定一种单钥加密算法 $E(\cdot)$ 及相应解密算法 $D(\cdot)$, 同样, $E(\cdot)$ 及 $D(\cdot)$ 为公开算法, 如 3DES.

1.2 签发数字证书

(1) 采用离线法, 由 Root充当 CA 为 A 及 S_i 签发数字证书. A 的数字证书为: $C_A = (ID_A, R, pub_A, t_A, s_A)$. 其中, ID_A 是 A 的 IP地址; R 是 Root的名字; pub_A 为 A 的公开密钥; t_A 为有效期; s_A 是 Root对 (ID_A, R, pub_A, t_A) 的数字签名: $s_A = sig_R(ID_A, R, pub_A, t_A) \quad [h(ID_A, R, pub_A, t_A)]^d_R \bmod n_R$.

(2) S_i 的数字证书为: $C_i = (ID_i, R, pub_i, t_i, s_i)$. 其中, ID_i 为 S_i 的 IP地址; pub_i 为 S_i 的公开密钥; t_i 为有效期; s_i 为 Root对 (ID_i, R, pub_i, t_i) 的数字签名: $s_i = sig_R(ID_i, R, pub_i, t_i) \quad [h(ID_i, R, pub_i, t_i)]^d_R \bmod n_R$. 为了抗击 UKS(Unknown Key Share) 攻击, Root为 S_i 颁发数字证书时, 必须保证 S_i 身份的可靠性并且最好保证 S_i 公钥互不相同.

(3) Root给自己签发一张数字证书: $C_R = (ID_R, R, (e_R, n_R), t_R, s_R)$. 其中, (e_R, n_R) 为 Root的 RSA公钥及模数; t_R 为有效期, s_R 为 R 对 $(ID_R, R, (e_R, n_R), t_R)$ 的数字签名: $s_R = sig_R(ID_R, R, (e_R, n_R), t_R)$ $[h(ID_R, R, (e_R, n_R), t_R)]^d_R \bmod n_R$.

将 C_A、C_R 安全传送到 A 并妥善保存, 将 C_i、C_R 安全传送到每一台 S 并妥善保管. 所有数字证书均属公开信息.

2 协议实现

用于口令同步的通用混合密码传输协议的工作过程如图2所示.

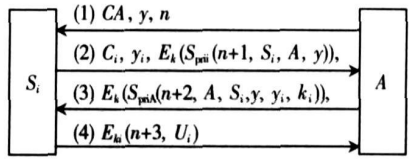

图 2 通用混合密码传输协议工作过程

Fig.2 The process of general hybrid cryptograph transfer protocol

(1) A 选定秘密参数 $x \quad GF(p)$, 在 GF(p) 中计算公开参数 $y \quad g^x \bmod p$, 向 S_i 发出 C_A, y, n, 其中 n 为随机数;

(2) 检验 C_A 上的 ID_A 值与实时连接所得到的 IP是否相等, 若不相符, 则中止与 A 的连接; 若相符, 则 S_i 验证收到的数字证书, 判断证书是否在有效期内, 读取当前系统时钟 d, 若 d t_A, 继续, 否则中止与 A 的连接;

取出证书中各元素, 计算 $h = (s_R)^e_R \bmod n_R$, $h = h(ID_A, R, pub_A, t_A,)$, 若 $h = h$, 则签名有效, 否则签名无效. 若 C_A 无效, 则中止与 A 的连接;

确认 C_A 有效且 IP地址合法后, 选定秘密参数 $x_i \quad GF(p)$, $(i = 1, 2, \ldots, I)$, 在 GF(p) 中计算公开参数 $y_i \quad g^{x_i} \bmod p$, 并计算主会话密钥 $K = y^{x_i} \quad g^{xx_i} \bmod p$. S_i 向 A 发送 $C_i, y_i, E_k(S_{prii}(n+1, S_i, A, y, y_i))$;

(3) A 收到 $(C_i, y_i, E_k(S_{prii}(n+1, S_i, A, y, y_i)))$ 后, 采用上述类似的方法检验 S_i 的 C_i、IP地址是否真实, 若其中一项不符合, 则中止与 S_i 的连接. 若真实, 则 A 计算主密钥 K: $K = y_i^x \quad g^{xix} \bmod p$;

A 用计算得到的主密钥解密 $E_k(S_{si}(2, S_i, A, y, y_i))$, 然后验证 S_i 的签名. 若验证成功, 则检查信息流的编号是否为 $n+1$, 信息流指向是否是 $S_i \quad A$, 若不是, 终止连接, 若是, 则选择随机数 k_i 作为会话密钥, 把 $(E_k(S_{priA}(3, A, S_i, y, y_i, k_i)), E_{ki}(\text{get}))$ 发送给 S_i, 其中 $E_k(S_{priA}(3, A, S_i, y, y_i, k_i))$ 使用主密钥 K 对用 A 的私钥签名的 $3, A, S_i, y, y_i, k_i$ 进行加密, $E_{ki}(\text{get})$ 使用会话密钥对命令字 get 进行加密;

(4) S_i 收到 $(E_k(S_{priA}(3, A, S_i, y, y_i, k_i)), E_{ki}(\text{get}))$ 后, 用主密钥求出 $S_{priA}(n+2, A, S_i, y, y_i, k_i)$, 验证签名并得到 $n+2, A, S_i, y, y_i, k_i$, 同上面一样, 检查信息流的编号和指向, 然后验证两个公开参数的值, 若前后一致, 则再用会话密钥 k_i 对 $E_{ki}(\text{get})$ 解密得命令字 get, 然后 S_i 取出 U_i 并把 $E_{ki}(U_i)$ 发向 A;

(5) A 收到 $E_{ki}(U_i)$ 后, 用前面生成的会话密钥 k_i 解密得到 S_i 上的数据库: $U_i = D_{ki}(E_{ki}(U_i))$.

经过上述 5个步骤, A 便可以安全地得到各个应用服务器的数据库 U_i.

3 协议分析

在设计协议时我们只能尽量避免错误, 但并不能保证没有错误. 为此, 应用 BAN逻辑[4] 进行如下形式

化分析. 为了符号表示的方便, 在以下协议分析中, A用于协商主密钥的公钥 y表示为 Y, 私钥为 x, A的用于验证签名的公开密钥 pub_A表示为 K_a, 用于签名的私钥 Spri_A表示为 K_a^{-1}; S_i用于协商主密钥的公钥 y_i表示为 Y_i, 私钥为 x_i, S_i的用于验证签名的公开密钥表示为 K_i, 用于签名的私钥 Spri_i表示为 K_i^{-1}; R的公钥 (e_r, n_r)表示为 K_r, R的私钥 d_r表示为 K_r^{-1}.

协议的目的是验证通信方的身份, 建立通信双方之间的共享主密钥 K, 并由 A产生会话密钥 K_s, 然后安全的传递给 S_i, 用 BAN类逻辑表示协议的目的如下:

$G1$: $A \mid \equiv S_i \overset{\sim}{\leftrightarrow} (*\ i,\ Y)$　　身份认证

$G2$: $A \mid \equiv A \overset{-}{\longrightarrow} S_i$　　安全密钥交换

$G3$: $A \mid \equiv A \overset{+}{\longrightarrow} S_i$　　密钥确认

$G4$: $A \mid \equiv \#K$　　密钥新鲜性

在以上的目标中, 没有包括 K_i应该满足的条件, 实际上, 如果 K能够满足以上的条件, 即能保证 K_i的新鲜性和 K_i传递的安全性. 所以我们只需将目光集中于上面提到的条件即可.

根据 BAN逻辑分析协议的步骤, 首先给出主体 A的初始化假设, S_i的初始化假设可对称得到. 关于主体 A的初始化假设为:

$P1$: $A \mid \equiv PK(R, K_r)$

$P2$: $A \mid \equiv SV([S_i, K_{Si}]_{Kr}^{-1}, K_r, S_t, K_{Si}))$

$A \mid \equiv SV([*\ i, Y]_{Ki}^{-1}, K_i^{-1}, (*\ i, Y))$

$P3$: $A \mid \equiv ((R \mid \sim PK(S_i, K_i)) \Rightarrow PK(S_t, K_i)$

$P4$: $A \mid \equiv PK(A, Y)$

$P5$: $A \mid \equiv \#Y$

$P6$: $A \mid \equiv A(Y, x)$

$P7$: $A \mid \equiv \{Y_s, (S_t, K_i, [S_i, K_i]_{Kt}^{-1}), \{[Y, Y_i]_{Ki}^{-1}\}_K\}$

$P8$: $A \mid \equiv A \mid \equiv \{*\ i, (S_t, K_t, [S_i, K_i]_{Kt}^{-1}), \{[*\ i, Y_i]_{Ki}^{-1}\}_K\}$

$P9$: $A \mid \equiv ((A \mid \equiv \{[*\ i, Y_i]_{Ki}^{-1}\}_K \Rightarrow PK(S_t, K_i)) \Rightarrow PK(A, Y)) \Rightarrow PK(S_t, *\ i))$

$P10$: $A \mid \equiv (A \mid \sim \{[*\ i, Y_i]_{Ki}^{-1}\}_K)$

$P11$: $(R \mid \sim (S_t, K_i)) \Rightarrow (R \mid \sim PK(S_t, K_i))$

根据 *BAN* 逻辑和初始化假设进行推理如下:

(1) $A \mid \equiv A \mid \equiv \{[*\ i, Y_i]_{Ki}^{-1}\}$　　从 $P8$、$Ax1$、$Ax7$、Nec、MP

(2) $A \mid \equiv R \mid \sim PK(S_t, K_i)$　　从 1、$P1$、$P2$、$P11$、$Ax1$、$Ax4$、Nec、MP

(3) $A \mid \equiv PK(S_t, K_i)$　　从 2、$P3$、$Ax1$、MP

(4) $A \mid \equiv A \mid \equiv \{[*\ i, Y_i]_{Ki}^{-1}\}_K$　　从 $P8$、$Ax1$、$Ax7$、Nec、MP

(5) $A \mid \equiv PK(S_t, *\ i)$　　从 3、4、$P4$、$P9$、$Ax1$、MP

(6) $A \mid \equiv A \overset{-}{\longrightarrow} S_i$　　从 5、$P4$、$Ax1$、$Ax5$、Nec、MP 这里 $K = F_0(Y_t, *\ i)$

(7) $A \mid \equiv AK$　　从 $P8$、$P6$、$Ax1$、$Ax10$、$Ax11$、$Ax12$、Nec、MP

(8) $A \mid \equiv A \overset{-}{\longrightarrow} S_i$　　从 6、7、$Ax1$、MP 和 $\overset{-}{\longrightarrow}$ 的定义 $(G2)$

(9) $A \mid \equiv \#K$　　从 $P5$、$Ax18$、MP $(K = F_0(Y, *\ i))$ $(G4)$

(10) $A \mid \equiv *\ \mathrm{confirm}_A(K)$　　从 4、9、$P10$、$Ax1$、MP 和 $*\ \mathrm{confirm}_A(K)$ 的定义

(11) $A \mid \equiv A \overset{+}{\longrightarrow} S_i$　　从 8、10、$Ax1$、MP 和 $A \overset{+}{\longrightarrow} S_i$的定义 $(G3)$

(12) $A \mid \equiv A \mid \equiv [*\ i, Y]_{Ki}^{-1}$　　从 4、7、$Ax1$、$Ax8$、Nec、MP

(13) $A \mid \equiv S_i \mid \sim (*\ i, Y)$　　从 3、12、$P2$、$Ax1$、$Ax4$、Nec、MP

(14) $A \mid \equiv S_i \mid \equiv \overset{\sim}{\leftrightarrow} (*\ i, Y)$　　从 13、$P5$、$Ax1$、$Ax19$、Nec、MP $(G1)$

至此, 我们得到了预期的目标 $G1$、$G2$、$G3$和 $G4$

4 结论

结合基于对称密钥的认证协议和基于公开密钥的认证协议, 我们设计了一种通用的认证协议 – 混合

密码协议. 本文给出了该协议的安全传送数据库部分的需求分析、算法设计、协议实现和 BAN 逻辑分析. 分析结果证明, 该协议能够达到预期的目标. 该混合密码协议存在一定的普适性, 可应用于任何两个能够提供数字证书的服务器之间交换共享密钥或共享数据. 例如: 单一登录服务中多个登录服务器之间数据的传输; web服务器之间共享密钥的传输; 以及 web认证中不同服务器之间的认证和数据交换, 如用户信息数据库的安全交换等.

[参考文献] (References)

[1] Menezes A, van Oorschot P, Vanstone S. Handbook of Applied Cryptography[M]. USA: CRC Press Inc, 1997.

[2] Blake-Wilson S, Meneses A. Unknown key-share attacks on the station-to-station (STS) protocol[D]. Canada: University of Waterloo, 1998.

[3] 韦卫, 王德杰, 张英, 等. 基于 SSL的安全 WWW 系统的研究与实现[J]. 计算机研究与发展, 1999, 36(5): 619-624.
Wei Wei, Wang Dejie, Zhang Ying et al. Study and implementation of a secure world wide web system based on secure sockets layer[J]. Journal of Computer Research and Development, 1999, 36(5): 619-624. (in Chinese)

[4] Burrow M, Abadi M, Needham R. A logic of authentication[J]. ACM Transactions on Computer Systems, 1990, 8(1): 18-36.

[责任编辑: 刘 健]