文章编号:1005－0523(2010)01－0042－05

广义椭圆曲线数字签名链口令认证方案

张利华,白二飞,杨秀青

(华东交通大学 电气与电子工程学院,江西 南昌 330013)

摘要:一次性口令是身份认证的重要技术。文章构造了一个基于椭圆曲线数字签名链的一次性口令认证和密钥协商方案。该方案使用了具有消息恢复功能、无须求逆的椭圆曲线数字签名算法,椭圆曲线认证密钥协商协议,密钥进化算法和椭圆曲线数字签名链等。方案有以下优点:服务器无需维护口令和验证列表;允许用户自主选择和更改口令,实现了双向认证;无需系统时钟同步和传输时延限制;能够抵抗重放攻击、离线字典攻击、中间人攻击和内部人攻击;具备口令错误敏感性和强安全修复性;生成的会话密钥具有新鲜性、机密性、已知密钥安全性和前向安全性。经对比,该方案具有更好的安全性能,适合强安全性需求的场合。

关 键 词:一次性口令;身份认证;密钥协商;数字签名链;椭圆曲线

中图分类号:TP393.08　　**文献标识码**:A

一次性口令认证技术[1－4]是信息系统安全中的重要技术,它能有效地防止网络上被动窃听者的攻击和重放攻击。目前,一次性口令的实现方式主要有四种:口令序列、挑战应答、时间同步、事件同步。常用的一次性口令的生成方式有:Token Card(硬件卡)、Soft Token(软件)、IC 卡。

S/KEY[5－7]认证方案是一次性口令的首次实现,但是该方案不能抵抗小数攻击和冒充攻击。SAS[8]方案可以克服 S/KEY 方案无法对抗小数攻击和需要重新初始化口令序列的缺点,且认证过程不需要进行大量的运算,减少了对系统资源的消耗,但是该方案存在无法对抗口令猜测攻击的缺陷。文献[9]等指出 SAS 口令认证方案在面对重放攻击和拒绝服务攻击时,仍然存在着安全漏洞。为此,Kamioka 和 Shimizu 提出了 SAS 口令认证的改进方案:使用智能卡的 SAS－1 和不使用智能卡的 SAS－2[10]。其中,SAS－2 认证方案具有以下特点:能有效抵御截取/重放攻击、伪造攻击和拒绝服务攻击等;在网上传输的都是用户的口令和其它计算因子通过异或运算后的认证数据,保密性较好;用户也不用重新进行初始化工作;在生成认证数据时所做 Hash 运算也大大减少,减少了对系统资源的占用;实现了双向认证。但是,SAS－2 认证方案需要在客户端保存随机数 N,若是需要用户记忆每次认证所需的随机数 N,则增加了用户的负担,给用户带来使用上的不便。

本文在研究典型一次性口令认证机制的基础上,针对其存在的安全缺陷,研究并构造了基于广义椭圆曲线数字签名链的一次性口令认证与密钥协商方案:EAKAS(Generalized elliptic curve digital signature chain based one time password authentication and key agreement scheme)。该方案利用了无须求逆且具有消息恢复功能的椭圆曲线数字签名算法、基于椭圆曲线的可认证的密钥协商协议[11]、密钥进化算法和椭圆曲线数字签名链。经分析比较,该方案有更好的安全性能。

1 基于广义椭圆曲线数字签名算法的数字签名链

文献[12]参照广义 ELGamal 签名方程构造方法给出了 18 种广义椭圆曲线签名算法,使用其中的不需要求逆、且具有消息恢复功能的广义椭圆曲线签名算法构造的数字签名链定义为:S 是一种椭圆曲线数字签名算法,V 为验证算法,d 和 Q 分别为用户的私钥和公钥,构造一对签名验证过程函数,$S_d(x)=y$ 和 $H_Q(x,y)$,定义如下。

收稿日期:2009－10－21

基金项目:国家自然科学基金项目(60271012);江西省教育厅科学技术研究项目:(赣教技字[2006]176 号)

作者简介:张利华(1972－),男,副教授,博士,主要研究方向为无线网络、信息安全。

$$S_d(x)=S_d(S_d(x));$$

$$H_Q(x,y)=\begin{cases}\text{Ture} & \text{if } V_Q(x)=\text{ture}\\ \text{False} & \text{if } V_Q(x)=\text{false}\end{cases}$$

每次签名利用上一次签名的结果,再进行签名,形成如下一个签名链:x, $S_d(x)$, $S_d^2(x)$, …, $S_d^n(x)$, $S_d^{n+1}(x)$, …,每次签名的结果做为一次性口令。

2　密钥进化算法

在密钥进化算法中,用户和服务器在一次身份认证和密钥协商过程采用的会话密钥是利用前一次的会话密钥,通过加密计算进化而来。在注册过程中用户 i 和服务器共享了初始密钥 k_0, N_0, N_j 均为随机数,$k_0=E_{N_0}(N_0||PW||H(ID))$,则在身份认证和密钥协商过程中采用的会话密钥为

第 1 次身份认证和密钥协商时,$k_1=E_{k_0}(k_0||H(ID_i)||N_1)$;

第 2 次身份认证和密钥协商时,$k_2=E_{k_1}(k_1||H(ID_i)||N_2)$;

第 $j(j\geqslant 1)$次身份认证和密钥协商时,$k_j=E_{k_{j-1}}(k_j-1||H(ID)||N_j)$。

因此,每次身份认证和密钥协商时,会话密钥都是不相同的,是以前面的会话密钥作为密钥,通过加密计算"进化"而来。由于随着 j 的不断增大,k_{ij}的位数会不断增加,在实际应用中取其前 n 位。

3　基于椭圆曲线数字签名链的一次性口令认证与密钥协商方案

EAKAS 类似于 S/KEY,采用挑战应答机制,整个方案分为注册阶段、身份认证和密钥协商阶段、会话密钥验证阶段和口令更改阶段。其中用户第一次注册、口令遗忘时执行注册阶段,用户需要更改口令时执行口令更改阶段,用户需要登陆主机时执行身份认证和密钥协商阶段、会话密钥验证阶段。

3.1　注册阶段

用户 U_i 需要输入用户标识符 ID_i 和用户口令 PW_i,客户端将这些数据进行计算后提交给服务器 AS 进行处理。

AS 生成如下参数:安全的 Hash 函数:$h(\cdot)$,选取安全的椭圆曲线参数为 $D=(q, FR, S, a, b, P, n, h)$ 生成公私钥对(d_s, Q_s),公私钥对秘密保存。U_i 是请求 AS 注册的第 i 个用户,利用同样的参数 $D=(q, FR, S, a, b, P, n, h)$生成公私钥对$(d_{U_i}, Q_{U_i})$,公私钥对秘密保存。通过安全信道向 AS 提出注册请求:提交其身份标识 ID_i、口令 PW_i 和公钥 Q_{U_i}。

Step 1　$AS\Rightarrow U_i$:$(h(\cdot), k_{i0})$。$\Rightarrow$:表示安全信道,$||$:连接符。AS 计算 $H(ID_i)$并查询认证数据库,若 ID_i 已经存在,则返回信息要求用户重新选择 ID_i,若该 ID_i 未被占用,则服务器生成随机数 N_{i0},计算 $k_{i0}=E_{N_{i0}}(N_{i0}||PW_i||H(ID_i))$,计算 $A_{i1}=E_{Q_S}(k_{i0})$, $A_{i2}=E_{Q_S}(Q_{U_i})$;

Step 2　$U_i\Rightarrow AS$:PW_{Ai0}。用户端计算 $A_{i3}=E_{PW_i}(k_{i0})$,利用给出的广义椭圆曲线签名算法,计算 $PW_{Ai0}=S_d(H(ID_i)||PW_i)$,并保存 $H(\cdot)$, A_{i3};

Step 3　AS 收到 PW_{Ai0}后,利用验证算法,验证签名的合法性,如果合法,允许注册。计算 $A_{i4}=E_{Q_S}(PW_{Ai0})$,保存 $H(ID_i)$, A_{i1}, A_{i2}, A_{i4}。

3.2　身份认证和密钥协商阶段

假设用户是第 $j+1$ 次登陆,此时用户拥有的数据包括 $H(ID_i)$、PW_i、其公私钥对,以及用其口令加密的上一次认证会话密钥值 $A_{i3}=E_{PW_i}(k_{ij})$。服务器拥有的数据包括其公私钥对、$H(ID_i)$、其公钥加密的用户公钥值 $A_{i2}=E_{Q_S}(Q_{U_i})$和其公钥加密的上一次认证会话密钥 $A_{i1}=E_{Q_S}(k_{ij})$、用其公钥加密的一次性口令值 $A_{i4}=E_{Q_S}(PW_{A_{ij}})$。

第 $j+1$ 次身份认证和密钥协商过程如下。

Step 1　$U_i\rightarrow AS$:$(H(ID_i), C_1)$。$\rightarrow$:表示不安全信道。当 U_i 登录 AS 时,输入 ID_i、PW_i,计算 $H(ID_i)$,解密获得 k_{ij},计算 $C_1=E_{k_{ij}}(H(ID_i))$;

Step 2 $AS\rightarrow U_i:(C_3)$。AS 收到$(H(ID_i),C_1)$后，利用 $H(ID_i)$，查询数据库获得 A_{i1}并用其私钥解密 A_{i1}，获得 k_{ij}，计算 $C_2=D_{k_{ij}}(H(ID_i))$，比较 $C_2^?=H(ID_i)$，如否拒绝登陆请求。如果相等，AS 生成随机数 N_{ij+1}，解密 A_{i4}得 $PW_{A_{ij}}$，计算 $k_{ij+1}=E_{k_{ij}}(k_{ij}||H(ID_i)||N_{ij+1}))$，$C_3=E_{k_{ij}}(k_{ij+1}||N_{ij+1}||PW_{A_{ij}})$；

Step 3 $U_i\rightarrow AS:(C_4)$。U_i 收到信息(C_3)后，解密 C_3，获得 N_{ij+1}、$PW_{A_{ij}}$，计算 $k'_{ij}=E_{k_{ij}}(k_{ij}||H(ID_i)||N_{ij+1})$，判断 $k_{ij+1}=k'_{ij+1}$，如果相等，服务器合法，完成对服务器的认证。用户利用给出的椭圆曲线数字签名算法计算 $PW_{A_{ij+1}}=S_{d_{U_i}}(PW_{A_{ij}})$，并将一次性口令串 $PW_{A_{ij+1}}$转化成整数 t'_{ij}，计算 $t_{ij}=t'_{ij}\bmod n$，选择随机数 $d_1\in[1,n-1]$，计算 $Q_U=(d_1+t_{ij})P$；计算 $C_4=E_{k_{ij+1}}(PW_{A_{ij+1}}||Q_{U_{ij}})$；

Step 4 $AS\rightarrow U_i:(C_5)$。AS 收到(C_4)后，解密获得 $PW_{A_{ij}}$，利用验证算法进行验证，验证通过，用户合法。选择随机数 $d_2\in[1,n-1]$，计算 $Q_{AS_{ij}}=(d_2+t_{ij})P$，$Y_{AS_{ij}}=Q_{U_{ij}}+(-t_{ij})P=d_1P$，$K_{AS_{ij}}=d_1d_2P$，$t_{ij}K_{AS_{ij}}=t_{ij}d_1d_2P$，计算 $C_5=E_{k_{ij+1}}(Q_{AS_{ij}}||t_{ij}K_{AS_{ij}})$。

3.3 会话密钥验证阶段

第 $j+1$ 次会话密钥验证过程如下。

Step 5 $U_i\rightarrow AS:(C_6)$。AS 收到(C_5)，计算 $X_{U_{ij}}=Q_{AS_{ij}}+(-t_{ij})P=d_2P$，$K_{U_{ij}}=d_1d_2P$，$t_{ij}K_{U_{ij}}=t_{ij}d_1d_2P$，$t_{ij}X_{U_{ij}}=t_{ij}d_2P$，判断 $t_{ij}K_{U_{ij}}^?=t_{ij}K_{AS_{ij}}$，如果通过，表明用户获得和服务器相同的会话密钥 $K_{US}=d_1X_{U_{ij}}=d_1d_2P$；计算 $C_6=E_{k_{j1}}(t_{ij}X_{U_{ij}})$。

Step 6 服务器收到(C_6)，验证 $t_{ij}X_{U_{ij}}^?=t_{ij}d_2P$，如果成功，表明服务器获得和用户同样的会话密钥 $K_{AS}=d_2Y_{AS_{ij}}=d_1d_2P$。服务器重新计算并更新 $A_{i1}=E_{Q_S}(k_{ij+1})$，$A_{i4}=E_{Q_S}(PW_{A_{ij+1}})$，并向用户发出认证成功的信息。

Step 7 用户收到认证成的信息后，重新计算并更新 $A_{i3}=E_{PW_i}(k_{ij+1})$。

3.4 口令更改阶段

当 U_i 发现自己的口令有泄漏的迹象或者为了增强安全性，需要修改口令时，在完成正常登陆并完成双向认证后可以将口令 PW_i 改为 PW_{new}。口令更改过程如下。

Step 1 $U_i\rightarrow AS:(H(ID_i),C_1)$。当 U_i 登录AS 时，输入 ID_i、PW_i，计算 $H(ID_i)$，解密获得 k_{ij}，计算 $C_1=E_{k_{ij}}(H(ID_i))$；

Step 2 $AS\rightarrow U_i:(C_3)$。AS 收到$(H(ID_i),C_1)$后，利用 $H(ID_i)$，查询数据库获得 A_{i1}并用其私钥解密 A_{i1}，获得 k_{ij}，计算 $C_2=D_{k_{ij}}(H(ID_i))$，比较 $C_2^?=H(ID_i)$，如否，拒绝登陆请求。如果相等，AS 生成随机数 N_{ij+1}，解密 A_{i4}得 $PW_{A_{ij}}$，计算 $k_{ij+1}=E_{k_{ij}}(k_{ij}||H(ID_i)||N_{ij+1})$，$C_3=E_{k_{ij}}(k_{ij+1}||N_{i1}||PW_{A_{ij}})$；

Step 3 $U_i\rightarrow AS:(C_4)$。U_i 收到信息(C_3)后，解密 C_3，获得 N_{ij+1}、$PW_{A_{ij}}$，计算 $k'_{ij+1}=E_{k_{ij}}(k_{ij}||H(ID_i)||N_{ij+1}))$，判断 $k'_{ij+1}^?=k_{ij+1}$，如果相等，服务器合法，完成对服务器的认证，用户输入新口令 PW_{new}。利用给出的椭圆曲线数字签名算法计算 $PW_{A_{ij+1}}=S_{dui}(PW_{A_{ij}}||PW_{new})$，计算 $C_4=E_{k_{ij+1}}(PW_{A_{ij+1}})$；

Step 4 AS 收到(C_4)后，解密获得 $PW_{A_{ij+1}}$，利用验证算法进行验证，验证通过，用户合法，此时计算并更新 $A_{i4}=E_{Q_S}(PW_{A_{ij+1}})$、$A_{i1}=E_{Q_S}(k_{ij+1})$，并向用户发出口令修改成功的信息。若不能通过，则拒绝口令修改过程，并将登陆错误次数加1(最大次数限定为3次)；

Step 5 如口令修改成功，用户收到信息后，用户计算并更新 $A_{i3new}=E_{PW_{new}}(k_{ij+1})$。

4 EAKAS 的安全性分析

命题1 EAKAS 协议未使用口令表，允许用户自主选择并更改口令，具备口令的用户友好性和主机的非透明性，能抵御离线字典攻击，实现了双向认证。

证明 EAKAS 协议未存储用户的口令，而且实现了用户和服务器的双向认证，交互的信息采用利用密钥进化算法生成的认证会话密钥加密，提供了口令更改机制，因此，命题1是成立的。

命题2 EAKAS 协议无须系统时钟同步及传输时延限制，能够抵御重放攻击。

证明　EAKAS 协议利用随机数替代时戳，无须系统时钟同步及传输时延限制，交互的信息利用一次性密钥进行了加密，在认证和会话密钥协商阶段的 Step 2 和 Step 3 均引入随机数，敌手不能有效实施重放攻击。

命题 3　EAKAS 协议能够抵御中间人攻击、内部人攻击，具备口令错误敏感性和强安全修复性。

证明　EAKAS 协议利用采用随机数参与运算进化生成的认证会话密钥加密交互信息，认证会话密钥具有一次性特征，即使敌手获得用户和服务器存储的信息，但是在用户和服务器私钥保密的情况下，即使同时用户的 PW 泄漏，敌手也不能获得有效的信息发起攻击，因而系统能够抵御中间人攻击、内部人攻击。

在用户的 PW 输入错误的情况下，用户不能获得正确的认证会话密钥，因而在认证和会话密钥协商阶段的 Step 2 验证不能通过，服务器拒绝登陆请求。

即使单个用户的信息泄漏，由于认证会话密钥采用的类似一次性的机制，不会影响到整个系统的安全性，不会影响其他用户的正常登陆。而且系统可以在身份认证和密钥协商阶段的 Step 2 和 Step 4 拒绝假冒用户登陆、在口令更改阶段 Step 2 和 Step 4 拒绝假冒用户更改口令，有效修复系统安全。

命题 4　EAKAS 协议生成的认证会话密钥、共享会话密钥是安全的，具有新鲜性、机密性、已知密钥安全性和前向安全性。

证明　认证会话密钥采用密钥进化算法生成，具有一次性特性；共享会话密钥和随机数有关，同样具有一次性特性，因此认证会话密钥和最终生成的共享会话密钥是安全的，具有新鲜性、机密性、已知密钥安全性和前向安全性。

EAKAS 协议与典型的一次性口令认证协议安全性比较如表 1 所示。其中，/：不满足；√：满足；C^1：双向认证；C^2：强安全修复性；C^3：交互信息机密性；C^4：抗小数攻击；C^5：抗假冒用户攻击；C^6：抗拒绝服务攻击；C^7：抗重放攻击；C^8：抗口令猜测攻击；C^9：抗中间人攻击；C^{10}：抗伪主机攻击；C^{11}：生成会话密钥。

表 1　EAKAS 协议与典型的一次性口令认证协议安全性比较

方案	对比项										
	C^1	C^2	C^3	C^4	C^5	C^6	C^7	C^8	C^9	C^{10}	C^{11}
S/KEY	/	/	/	/	/	/	/	/	/	/	/
SAS	/	/	/	√	/	/	/	/	/	/	/
SAS－2	√	/	√	√	/	√	√	√	/	√	/
EAKAS	√	√	√	√	√	√	√	√	√	√	√

5　EAKAS 的效率分析

EAKAS 虽然采用无需求逆的椭圆曲线数字签名算法来生成一次性口令，但是由于使用了数字签名算法生成一次性口令、利用密钥进化算法生成认证会话密钥、在认证过程使用加密算法加密交互信息、最后生成的共享会话密钥引入确认环节，同时需要在客户端和服务器端存储相应的数据，整个协议的资源消耗（存储需求、运算需求、交互次数等）比一般认证协议要大，但是考虑到硬件技术的快速发展，资源消耗的问题在用户数量规模不大的情况下不会对系统性能造成较大的影响。因此，在首要考虑系统安全强度的情况下，在一些安全性要求较高的应用场合比如存储保密级别较高的数据的应用系统中，EAKAS 是一个比较合适的解决方案。需要指出的是，为进一步提高系统的安全性，可以将客户端存储的信息存放在智能卡、USB Key 等存储介质上。

EAKAS 协议与典型的一次性口令认证协议效率（登录及认证阶段）比较如表 2 所示，其中：P^1，客户端计算开销；P^2，服务器端计算开销；P^3，信息交换次数；Hash，Hash 运算；Mul，乘积运算；En，加密运算；De，解密运算；Mod，模运算；Sign，签名运算；Ver，验证运算；Ran，生成随机数次数。

表 2　EAKAS 协议与典型的一次性口令认证协议效率（登录及认证阶段）比较

方案	对比项		
	P^1	P^2	P^3
SAS	$^2Ran+^6Hash$	^1Hash	3
SAS－2	$^5Hash+^2Ran$	^2Hash	4
EAKAS	$^1Hash+^2Sign+^1Mod+^1Ran+^1Mul+^2De+^4En$	$^2Ver+^3Ran+^4De+^8En+^5Mul+^1Hash$	7

6 小结

EAKAS方案采用低开销的、非求逆、具有消息恢复功能的椭圆曲线数字签名算法，在一定程度上解决了使用公钥体制带来的资源消耗过大的问题；采用经过密钥进化算法生成的认证会话密钥代替共享的系统密钥来加密交互的信息，可以保证认证过程中交互信息的机密性；利用经过椭圆曲线签名算法生成的重复签名做为一次性口令，确保其安全性。经过安全分析，该方案允许用户自主选择并更改口令，能抵御离线字典攻击，实现了双向认证；无须系统时钟同步及传输时延限制，能够抵御重放攻击；能够抵御中间人攻击、内部人攻击，具备口令错误敏感性和强安全修复性，其适合于强安全性要求的场合。

参考文献：

[1] LAMPORT L·Password authentication with insecure communication[J]·Communications of the ACM,1981,24(11):770－772.
[2] 王滨，张远洋·一次性口令身份认证方案的分析与改进[J]·计算机工程，2006,32(14):149－150.
[3] 袁丁，凡平志·一个安全的远程动态口令鉴别方案[J]·四川大学学报：自然科学版，2002,39(2):228－232.
[4] 郑海龙，刘建伟·一次性口令认证机制的分析与研究[J]·信息安全与通信保密，2008,20(11):64－66.
[5] HALLER N M·The S/Key(TM) one－time password system[C]·Proc·Internet Society Symposium on Network and Distributed System Security·San Diego,CA,USA:ISSNDSS,1994:151－158.
[6] 王世卿，雷根平，等·S/Key协议的分析与改进[J]·微计算机信息，2008,24(36):60－61,173.
[7] 李世平，李凤霞，战守义·S/KEY认证系统的安全缺陷及改进[J]·计算机工程，2003,29(20):18－19.
[8] SANDIRIGAMA M,SHIMIZU A,NODA M T·Simple and secure password authentication protocol (SAS)[J]·IEICE Trans Commun,2002,E83－B (6):1 363－1 365.
[9] LIN C L,SUN H M,HWANG T·Attacks and solutions on strong－password authentication[J]·IEICE Trans on Commun,2001,E84－3 (9):2 622－2 627.
[10] TSUJI T,KAMIOKA T,SHIMIZU A·Simple and secure password authentication protocol,ver·2(SAS－2),OIS2002－30[R]·IEICE Technical Report,2002,102(314):7－11.
[11] 隋爱芬，杨义先，钮心忻，等·基于椭圆曲线密码的可认证密钥协商协议的研究[J]·北京邮电大学学报，2004,27(3):28－32.
[12] HAN L,GONG G·Euiptic Curve Digital Signatures and Accessories[C]·Hongkong:Proceeding of the Internatcinal Workshop on Gryptographic Techniques & E－commerce,1999:126－131.

Password Authentication Scheme of Generalized Elliptic Curve Digital Signature Chain

Zhang Lihua, Bai Erfei, Yang Xiuqing

(School of Electrical and Electronic Engineering, East China Jiaotong University, Nanchang 330013, China)

Abstract: One time password is an important technology of the user authentication. In this paper, we develop a novel elliptic curve digital signature chain based on one time password authentication and key agreement scheme (EAKAS). The scheme uses many mechanisms such as elliptic curve digital signature algorithm that can recover message and has no inversion, elliptic curve based on authenticated key agreement protocol, key evolutionary algorithm, and elliptic curve digital signature chain, etc. The scheme has the following merits: there is no need for any password or verification table in the server; users can choose or change password freely and achieves mutual authentication; it has no system clock synchronization and no transmission delay constraint; it can resist replay attacks, man－in－the－middle attack, off－line dictionary attack and insider attack; it has the feature of password error sensitivity and strong security restoration; the session keys in proposed scheme have the feature of freshness, confidentiality, known key security and forward security. By comparison, the scheme has better security and is well suited to occasion which requires strong security.

Key words: one time password; user authentication; key agreement; digital signature chain; elliptic curve

（责任编辑 王全金）