计算机科学 2002Vol. 29№. 8

一个基于 ECC 的双向认证协议*)

A Mutual Authentication Protocal based on Elliptic Curve Cryptography

张险峰　秦志光　刘锦德

(电子科技大学微机所　成都 610054)

Abstract　Identity authentication is important in network security. In this paper, a mutual authentication protocol based on Elliptic Curve Cryptography is designed and analysed. This protocol enables two entities to authenticate each other on computer network, defends any cheat, relay attack.

Keywords　Network security , Authentication, Mutual authentication, Elliptic curve cryptography(ECC)

1. 引言

身份认证是网络安全技术的一个重要方面,身份认证机制限制非法用户访问网络资源,能够防止假冒、篡改、否认等攻击,确保用户的身份,是其他安全机制的基础。双向身份认证是指通信双方需要互相认证鉴别各自的身份[1,2]。双向认证的典型方案是 Needham-Schroeder 协议。常见的认证协议还有分布认证安全服务(DASS)协议、ITU-T X.509 认证协议等。

上述基于公钥密码体制的身份认证系统的安全,从根本上是依靠所采用的公钥密码算法的安全强度。鉴于椭圆曲线密码体制(Elliptic Curve Cryptography,ECC)在安全强度方面特有的优越性,本文提出了一个基于椭圆曲线密码体制的双向身份认证协议。其安全依靠在椭圆曲线群里计算椭圆曲线离散对数的困难性,它将会在无线通信或其他网络环境里有着广阔的应用前景[3~7]。

2. 协议涉及的基本原理

ECC 涉及的主要运算是椭圆曲线群中几何点的运算。这些点是定义在有限域上椭圆曲线方程的解集中的元素。有限域一般采用 GF(p)和 GF(2^k)。我们先定义术语系统[5]:

1)标量:在 GF(p)和 GF(2^k)中的一个元素。一般用小写字母表示。

2)标量加法:两个或更多的标量相加可获得另外一个标量。在 GF(p)情形下,就是模 p 的普通整数加法。而采用 GF(2^k)时,等价于一个模度为 k 的不可约多项式的多项式加法,此不可约多项式为 GF(2^k)的生成多项式。

3)标量乘法:两个或两个以上的标量相乘可获得另一个标量。在 GF(p)情形下,就是模 p 的普通整数乘法。而采用 GF(2^k)时,等价于模度为 k 的生成多项式的多项式乘法。

4)标量逆元:在 GF(p)或 GF(2^k)中元素 a 的乘法逆元表示为:a^{-1},满足 $a\,a^{-1}=1$。乘法逆元可通过费尔马方法或欧几里德方法来计算。

5)点:满足椭圆曲线方程的一个有序标量对称为一个点。常用大写字母来表示。点也常用其坐标来表示,如:点 P 可表示成 $P=(x,y)$,x,y 为有限域中元素。进一步,点 P 的 x,y 坐标可分别表示成 P_x 和 P_y。

6)点加:这是一种通过一系列规则可在一给定的曲线上由两点 P、Q 得到第三个点 R 的方法。表示为:R=P+Q,它与标量加法是不同的。

7)点乘:一个椭圆曲线点 P 和一整数 e 的乘积,可表示为 $e\times P$,它等价于 e 个 P 相加,其结果产生了曲线上另外一点。

8)信息摘要函数:把一个长信息压缩成长为 128 或 160 比特的函数。两个被广泛应用并被标准化的信息摘要函数是 MD5 和 SHA。用 H(M)表示信息 M 的信息摘要。出于有效性的考虑,数据签名函数一般把 H(M)作为输入,而不直接对 M 进行操作。两信息 M_1 和 M_2 连接的信息摘要表示成 $H(M_1, M_2)$。

2.1　基于公钥密码技术的认证机制(图 0)

基于密码的认证机制的基本原理是:使验证者信服声称者是其所声称的,因为仅有声称者知道某一秘密密钥。具体到公钥密码技术,声称者使用他的私钥签署某一消息,验证者使用声称者的公钥检查签名。如果签名能被正确地检查,那么验证者相信声称者是其本人。一般,消息中包含了一个非重复值以抵抗重发攻击[2]。

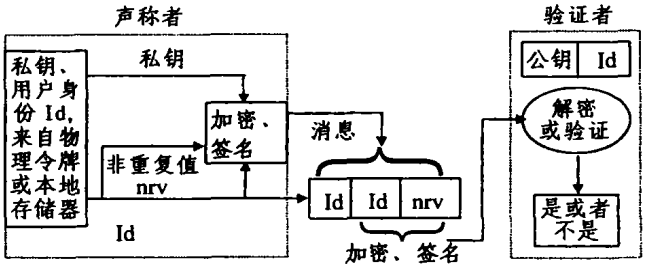

图 0　简化的基于公钥密码技术的认证机制

2.2　椭圆曲线数字签名算法(ECDSA)[4,6]

ECDSA 是本协议实现的核心,它是数字签名算法在椭圆曲线上的模拟。目前,ECDSA 已经由 ANSI XF1 和 IEEE P1363 标准化委员会制订了标准。ECDSA 的过程如下:

ECDSA 密钥产生　用户 A 按以下步骤:

1)选择一个定义在 GF(p)上的椭圆曲线 E,要求 E 上点的数目能被一大素数 n 整除。

2)选一点 P∈E,其阶为 n。

3)随机选取整数 d,d∈[1,n−1]。

*)本课题得到国防预研基金的资助。张险峰　博士生,主要研究方向:网络安全技术。秦志光　教授,主要研究方向:网络安全、电子商务。刘锦德　教授,博导,主要研究方向:开放系统及其安全、中间件技术。

4)计算 $Q=d\times P$。

5)A 的公钥是(E,P,n,Q);A 的私钥是 d。

ECDSA 签名的产生 用户 A 按如下步骤对信息 m 进行签名:

1)随机选取整数 k,$k\in[1,n-1]$。

2)计算 $k\times P=(x_1,y_1)$,且 $r=x_1 \bmod n$。(若 $x_1\in GF(2^t)$,则把 x_1 看作一二进制数。)若 $r=0$,则转步骤 1)。

3)计算 $k^{-1} \bmod n$。

4)计算 $s=k^{-1}\{H(m)+dr\} \bmod n$,这里 H 为安全杂凑算法 SHA-1。若 $s=0$,则转步骤 1)。

5)用户 A 对消息 M 的数字签名为整数对(r,s)。

ECDSA 签名验证 用户 B 采用以下步骤来验证用户 A 对 m 的签名:

1)获得 A 的公钥(E、P、n、Q),验证 r 和 s 都是区间[1,n-1]上的整数;

2)计算 $w=s^{-1} \bmod n$ 和 H(m);

3)计算 $u_1=H(m)w \bmod n$ 和 $u_2=rw \bmod n$;

4)计算 $u_1P+u_2Q=(x_0,y_0)$ 和 $v=x_0 \bmod n$;

5)若 $v=r$ 则接受签名;反之拒绝。

3. 双向认证协议的设计

要实现基于公钥密码体制的认证机制,前提条件是各个认证实体需拥有自己的数字证书。数字证书是由一个权威机构—CA(Certification Authority)颁发。证书包含有:由 CA 提供给请求方的临时身份、请求方的公钥、证书的有效期限、CA 用自己的私钥对前几项内容连接成的二进制串运算得到的数字签名。一个特定用户的身份通过其持有的证书和他的公钥进行绑定。用户的证书是在首次和服务器预定服务时获得的。

为了协议描述的方便,假定两个需认证的实体为用户和服务器。假定选取的椭圆曲线 E、基点 P 的定义同 2.2 节。服务器的私钥为 d_s,公钥为 Q_s;用户的私钥为 d_u,公钥为 Q_u;CA 的私钥为 d_{ca},公钥为 Q_{ca}。有 $Q_s=d_s\times P$;$Q_u=d_u\times P$;$Q_{ca}=d_{ca}\times P$。

3.1 服务器和用户的初始化

为获得一证书,服务器通过一安全、可信的信道把其公钥 Q_s 和其用户身份传送到 CA。CA 用其私钥对服务器公钥 Q_s、临时身份 I_S、证书的有效期 t,连接成的二进制串的信息摘要进行数字签名。然后,CA 按以下步骤(图 1)将已签的信息传送到服务器。

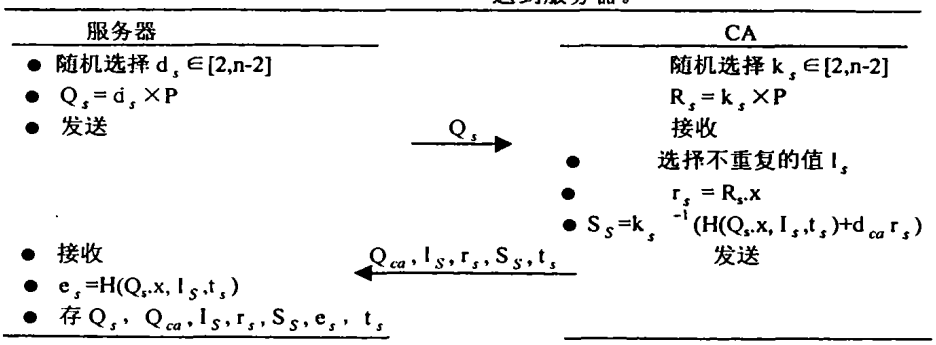

图 1 服务器初始化

通过执行以下(图 2)与服务器初始化几乎完全类似的步骤,用户可从 CA 获得其证书。

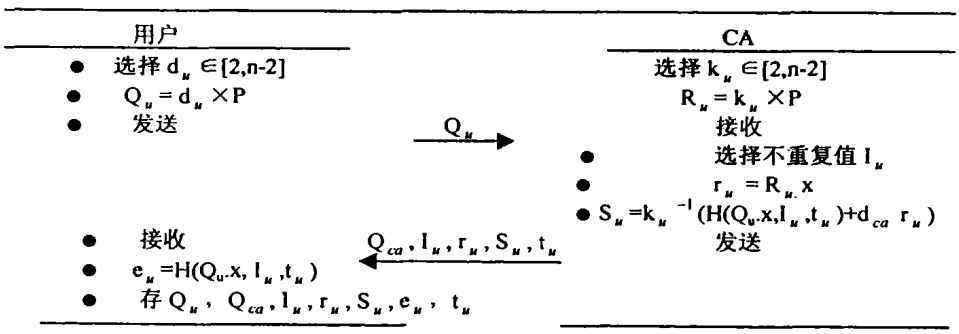

图 2 用户初始化

服务器和用户的证书分别表示为整数对:(r_s,s_s)和(r_u,s_u)。

3.2 用户和服务器间的相互认证

在实际的应用中,用户和服务器间的相互认证和密钥协定协议需要实时执行。图 3 中给出了一个将认证和密钥交换协议相结合的协议。协议中的数据采用对称密钥加密算法来实现保密。可采用传统的分组密码(DES,3DES,IDEA,RC5)或流密码(RC4,SEAL)并运用密码块链方式(CBC)来进行加密。用秘密钥 K 来对明方 M 和密文 C 进行加密和解密操作,分别表示为:$C:=E(K,M)$和$M:=D(K,C)$。

4. 协议的分析

为防重放攻击,安全协议一般采用非重复值的方法。本协议采取预先从验证者发送的随机值作为非重复值,验证者的责任是确保同一随机值在规定的时间内不被重复使用。另外两种使用非重复值的方法是:(1)在声称者和验证者之间保持序列号,这要求对每个验证者需保存每个声称者的状态信息。每个声称者和每个验证者都需保存所使用的同步状态信息。

在开放系统环境里，它加重了管理负担。(2)时戳。时戳假定逻辑上联系的声称者和验证者拥有一个共同的时间基准点，但在开放系统中，通信双方的时钟要保持严格的同步是困难的。而本协议所使用的方法避免了在声称者和验证者之间保持序列号而对系统增加的管理问题和时间戳的脆弱性，只是增加了附加的协议消息。

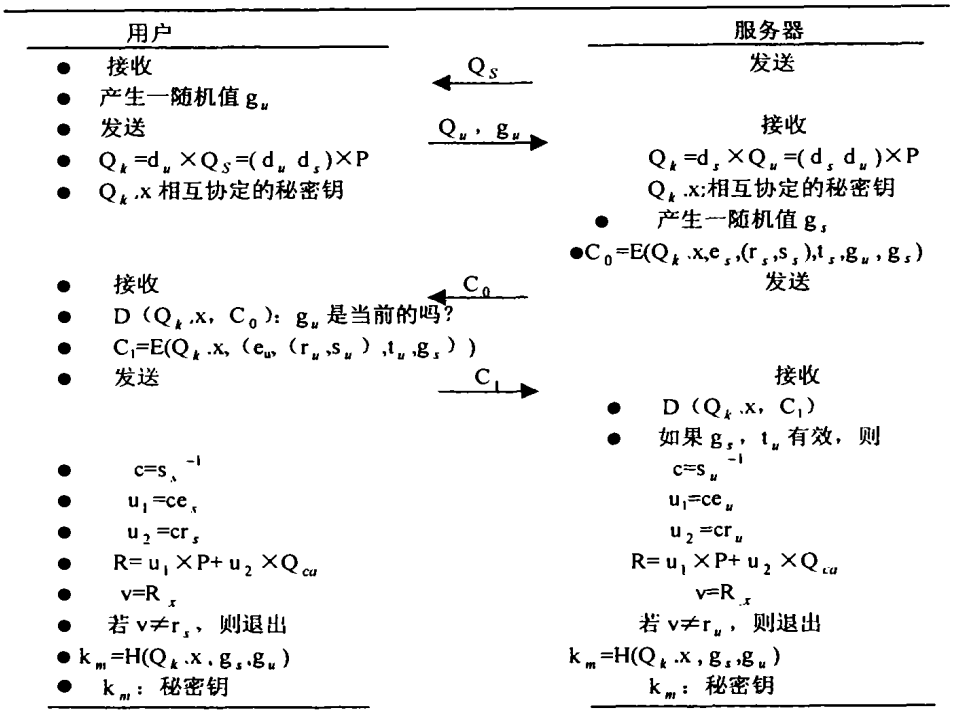

图 3　相互认证和密钥协定

根据协议，只要有用户或服务器请求服务，就立即进行一次密钥交换。被请求方将向正请求服务的一方发送一随机的询问。一旦双方得到对方的公钥，他们可立即产生一共享的秘密钥 $Q_k.x$。

服务器对证书的 e_s、(r_s,s_s)、证书有效日期 t_s、随机数 g_s 的连接二进制串进行加密，这里的 g_s 可用来获得最终相互通信的密钥。对证书内容加密可防止窃听者窃听证书，也可在一定程度上防止欺骗攻击。由于证书一般不太长(量级一般为 kbit)，而对称加密速度又非常快，所以加密时间很短。

加密的信息 C_0 传送到用户后，用户用 $Q_k.x$ 对 C_0 解密从而获得服务器的证书、随机数 g_s 和自己发出的询问 g_u。从服务器方获得开始发出的询问值证实了信息确为刚才发出从而防止了重放攻击。用户立即对它的证书 e_u、(r_u,s_u)、证书有效日期 t_u、随机数 g_s 的连接结果进行加密，加密结果记为 C_1 并发送给服务器。

接着，用户检查服务器证书的有效性，如果证书无效则退出通信。另一方面，服务器解密 C_1 并且检查 g_s 和时间证书是否有效。若无效则退出。这种机制，特别是 g_s 的应用，既防止了用户进行欺骗攻击，也避免了不必要的计算开销。

最后，服务器检查证书的有效性并根据结果决定是否提供服务。在这里，可以考虑预先产生多个随机值，以在执行协议时节省时间。当然，这是以增加协议的存储要求为代价的。

当双方结束认证过程后，用户和服务器就可在已建立的信道上开始通信了。虽然他们已经有了一共享的秘密钥 $Q_k.x$，但在他们证书有效期限内该值不能重用。因此，需要增加一新的密钥交换步骤以便在每次会话中协商产生用于通信的秘密钥。然而，执行另外的密钥协定过程会增加系统开销。我们可用协议中已产生的双方都已知的 g_s 和 g_u 来产生一新的秘密钥。服务器和用户只需都执行一信息摘要操作来获得一新的秘密钥 k_m。通过 k_m 双方可加解密通过信道的数据。

结论　本文详细描述了一个基于 ECC 的双向认证协议。目前，ECC 被认为能比其他公钥体制提供更好的加密强度、更快的执行速度和更小的密钥长度[3,6]，所以该协议能与其他协议相比，能够减少认证机制中的计算量和通信量，同时又提供较高的安全性能。

1997 年以来，ECC 及其安全性分析引起了密码学家及各界的极大关注与重视，现已形成了研究热点，但直到目前，在密码分析方面仍未取得实质性进展，因而大多数密码学家对这种密码体制的强度及应用前景越来越抱乐观态度，SET 协议的制定者已把它作为下一代 SET 协议中缺省的公钥密码算法。有理由相信，ECC 的优越性能使它必将取代 RSA，成为通用的公钥加密算法。所以，基于 ECC 的安全应用是一个非常值得研究的课题。

参 考 文 献

1　Schneier B. Applied Cryptography. New York: Jone Wiley & Sons, Inc., 1994. 125～137

2　冯登国. 计算机通信网络安全. 清华大学出版社，2001. 56～66

3　张险峰，秦志光，刘锦德. 椭圆曲线加密体制的性能分析. 电子科技大学学报，2001

4　IEEE P1363. Standard Specifications for Public Key Cryptography. Draft Version 13, 1999

5　Menezes A J. Elliptic Curve Public Key Cryptosystems. USA: Kluwer Academic Publishers, 1993

6　Certicom Corporation Whitepaper. Canada: Certicom Corporation, 1997

7　Rhce M Y. Cryptography and Secure Communications. USA: McGraw-Hill Co, 1994 Wayne Patterson. Mathematical Cryptology for Computer Scientists and Mathematicians. USA: Rowman&Dittlefield, 1987