对一种自证明签名方案的攻击和改进

李新国[1,2], 葛建华[1], 赵春明[1]
(1. 西安电子科技大学 ISN 国家重点实验室, 陕西西安 710071; 2. 解放军洛阳外国语学院数学教研室, 河南洛阳 471003)

摘 要: 本文对一种自证明签名方案实施了伪造攻击. 在这种攻击下 CA 通过改变数字签名中的相关参数, 可以对其用户的任意消息伪造自证明签名. 本文提出了能抵抗该攻击的改进方案, 分析表明改进方案和 Schnorr 签名具有相同的安全性.

关键词: 数字签名; 自证明签名; 伪造攻击
中图分类号: TP309 文献标识码: A 文章编号: 0372-2112 (2004) 08-1364-03

Improved Scheme of the Forgeable Self-Certified Signature

LI Xin-guo[1,2], GE Jian-hua[1], ZHAO Chun-ming[1]
(1. National Key Lab of ISN, Xidian University, Xi. an, Shaanxi 710071, China;
2. Dept. of Mathematics, PLA Foreign Language Institute, Luoyang, Henan 471003, China)

Abstract: A forgery attack is presented on the self-certified signature scheme. The results show that CA can forge a valid self-certified signature of his client on an arbitrary message by tampering with some parameters of the signature. An improved scheme is proposed to resist this kind of forgery attack. Analyses show that the improved scheme has the same security features with Schnorr signature.

Key words: digital signature; self-certified signature; forgery attack

1 引言

数字签名可以保证所签消息的完整性、认证性和不可拒认性, 这些功能的实现一般都假定有一个可信任的第三方(TTP). 在 X. 509[1] 中, CA(Certification authority) 充当可信第三方的角色为其用户颁发数字公钥证书. 证书中有用户公开密钥、用户身份、序列号、CA 身份等内容, 以及 CA 对上述所有内容的数字签名. 证书持有者利用与其公钥证书中的公开密钥对应的私钥对消息进行签名. 验证人验证签名人对消息的签名时, 首先要利用签名人公钥证书中的公钥验证对消息的签名, 接下来还必须验证 CA 对签名人公钥证书的签名. 所以验证过程实际上包括两次签名验证. 另外, 验证人还不得不通过 CRL 或者其他途径[2]获取签名人公钥证书的吊销信息.

文献[3]的作者指出:/ 站在验证人的角度来看, 验证两个独立主体的数字签名是一种负担0. 他们提出了自证明签名(Self-certified signature)的概念和方案. 在自证明签名方案(SCS)中, 签名人用他的长期签名私钥和证书信息生成自证明签名公钥私钥对, 验证人只需验证签名人对消息的自证明签名, 从而省去了一次对证书的签名验证过程. 自证明签名所取得的效率优势在于减少了需验证的签名的数量, 而且 SCS 的使用并不影响现有 PKI 的基本结构, 这也是其具有应用前景的原因之一. 本文的主要结果是构造了对原始方案[3]的一种伪造攻击, 并提出了一种改进方案来避免这种攻击.

2 自证明签名

自证明签名使用 Schnorr 签名方案[4]作为基础. 我们首先简要地对 Schnorr 签名作一描述. 设 p 和 q 是两个大素数, 满足 q|(p-1); g 是群 Z_p^* 的一个阶为 q 的乘法子群的生成元. H 表示抗碰撞 Hash 函数. 假设签名人的公钥是 y, 对应的私钥是 x, $y \equiv g^x \bmod p$. 当签名消息 m 时, 签名人随机地取 $k \in_R Z_q^*$ 并计算:

$$r \equiv g^k \pmod p \text{ 和 } s \equiv x\#H(m, r) + k \pmod q$$

那么(r, s)就是对消息 m 的有效签名. 有效性是由式子 $g^s \equiv y^{H(m,r)}\#r \pmod p$ 保证的. Schnorr 签名方案已经被证明在随机问答器模型下是安全的[5].

本节的以下部分是对原自证明签名方案的描述. 假设签名人 S 有一个长期的公私钥对(x_0, y_0), 其中 $y_0 \equiv g^{x_0} \pmod p$. 签名人还有一个由 CA 颁发的数字证书. CA 的公私钥对和签名人的公私钥对使用同样的 Schnorr 公钥系统, 即系统参数 p、q 和 g 是一样的. 假设 CA 的公私钥对是(x_{CA}, y_{CA}), 其中 $y_{CA} \equiv g^{x_{CA}} \pmod p$. 由 CA 准备的用户证书信息是 CI_s, CI_s 由序列号、用户长期公钥 y_0、用户身份、CA 的身份等信息组成.

收稿日期: 2003-06-09; 修回日期: 2004-03-26
基金项目: 国家教育部高校骨干教师资助(No. 2000-22001)

颁发关于公钥的证书时, CA 随机取 $k_c \in_R Z_q^*$, 则用户 S(签名人)的证书就是:

$$\{CI_s, (r_c, s_c)\} = \{CI_s, (g^{k_c} (\mathrm{mod}\ p), x_{CA} \# H(CI_s, r_c) + k_c (\mathrm{mod}\ q))\}$$

S 可以通过下式验证其证书的有效性: $g^{s_c} \stackrel{?}{=} y_{CA}^{H(CI_s, r_c)} \# r_c (\mathrm{mod}\ p)$.

接下来可以分密钥生成、签名和验证三部分来描述对消息 m 的自证明签名, 在此期间签名人利用了他的公钥证书信息.

密钥生成: 签名人 S 基于其长期公私钥对(x_0, y_0)和 CA 颁发的证书计算出他的 SCS 密钥对(x, y), 其中, $x = x_0 + s_c (\mathrm{mod}\ q)$, $y = y_0 \# y_{CA}^{H(CI_s, r_c)} \# r_c (\mathrm{mod}\ p)$.

签名: 使用 SCS 签名密钥 x, S 计算出他对消息 m 的自证明签名 R= (r, s), 方法如下: S 随机取$k \in_R Z_q^*$, 并计算:

$r = g^k (\mathrm{mod}\ p)$, $s = x \# H(m + CI_s + r_c, r) + k (\mathrm{mod}\ q)$.

则$\{(r, s), CI_s, r_c\}$就是验证人收到的 S 关于消息 m 的签名.

验证: 验证人通过以下三步验证自证明签名$\{(r, s), CI_s, r_c\}$的有效性:

(1)计算出签名人的 SCS 公钥, $y = y_0 \# y_{CA}^{H(CI_s, r_c)} \# r_c (\mathrm{mod}\ p)$;

(2) 用上一步计算出的 y 验证签名(r, s), $g^s \stackrel{?}{=} y^{H(m + CI_s + r_c, r)} \# r (\mathrm{mod}\ p)$;

(3)检查签名人的长期公钥 y_0 是否与 CI_s 中的一致(这一步仅仅是字符检查, 并非签名验证).

以上只是对基于离散对数问题的 SCS 的一般实现的一个描述. 在文[3]中, 作者进一步将 SCS 的一般实现进行了改进和扩展. 因为本文的攻击方法同样适用于那里的改进和扩展方案, 为了简洁, 这里略去对改进和扩展方案的描述. 有兴趣的读者可以参看文献[3], 并将我们的攻击方法加以套用.

3 对 SCS 的伪造攻击

我们首先对上一节中 SCS 签名的使用环境作进一步的分析. 在签名人一方, 他有 CA 颁发的公钥证书和相应的长期签名私钥, 签名人利用证书和长期签名私钥私下计算出 SCS 签名私钥和公钥, SCS 公钥无需公开; 在验证人一方, 他要通过某种途径获取签名人的公钥证书但并不验证 CA 在该证书上的签名. SCS 签名的效率优势在上一节的描述中并没有体现出来. 如果将 SCS 签名扩展到 PKI 中需要许多认证的信息(包括证书吊销列表 CRL 和从根 CA 到签名人的证书链)的场合, 其优势将非常明显[3].

本节, 我们指出 CA 可以产生伪造的 SCS 签名. 注意到, 在验证人的验证过程中, 签名人的公钥证书$\{CI_s, (r_c, s_c)\}$自始至终都没有用 CA 的公开密钥 y_{CA} 验证过, 所以上一节的自证明签名$\{(r, s), CI_s, r_c\}$中的所有内容都可能被恶意的攻击者窜改或伪造. 然而, 由于 SCS 签名实际上和 Schnorr 签名具有很多的相同之处, 我们发现除了 CA 以外的任何攻击者都不能伪造出一个有效的 SCS 签名(在随机问答器模型下). 通过以下步骤, CA 可以伪造出签名人 S 对任何消息 m 的自证明签名.

) 对于消息 m, CA 随机取 $k \in_R Z_q^*$ 并计算 $r = g^k (\mathrm{mod}\ p)$;

) CA 可以方便地计算出 $r_c = y_0^{-1} (\mathrm{mod}\ p)$, 因为 y_0 是乘法循环群中的一个公开元素;

) CA 利用他的签名私钥 x_{CA} 和他想伪造的签名人的证书信息计算:

$s = x_{CA} \# H(m + CI_s + r_c, r) \# H(CI_s, r_c) + k (\mathrm{mod}\ q)$

CA 将伪造的签名$\{(r, s), CI_s, r_c\}$呈给某个验证人.

验证人按照上一节介绍的验证步骤验证签名$\{(r, s), CI_s, r_c\}$如下:

(1)计算出签名人的 SCS 公钥,

$y = y_0 \# y_{CA}^{H(CI_s, r_c)} \# r_c = y_0 \# y_{CA}^{H(CI_s, r_c)} \# y_0^{-1} = y_{CA}^{H(CI_s, r_c)} (\mathrm{mod}\ p)$;

(2)验证方程 $g^s \stackrel{?}{=} y^{H(m + CI_s + r_c, r)} \# r (\mathrm{mod}\ p)$是否成立;

(3)检查签名人的长期公钥 y_0 是否与 CI_s 中的一致.

到此为止, 可以看到伪造的签名$\{(r, s), CI_s, r_c\}$的确是对消息 m 的一个有效的签名: 签名人的长期公钥 y_0 依然原封不动地置于 CI_s 中, 验证方程也是成立的. CA 甚至可以将 CI_s 中的证书的序列号加以改动, 使得验证人得不到关于签名人证书的正确的吊销信息, 除非证书中有关于证书内容的 Hash 值用以保证内容的完整性.

4 改进的 SCS 方案

在上一节中, CA 将 SCS 签名中的参数 r_c 置为 y_0^{-1}, 此时签名人的签名私钥就不再起任何作用, 这是导致签名伪造的根本原因. 本节, 我们将签名人的签名私钥 x 和相应的公钥稍微加以改变, 以使 CA 不再可以伪造出有效的 SCS 签名.

下面分密钥生成、签名和验证三部分来描述改进的方案.

密钥生成: 签名人 S 基于其长期公私钥(x_0, y_0)对和 CA 颁发的证书计算出他的 SCS 密钥对(x, y), 其中,

$x = x_0 H(CI_s, r_c) + s_c (\mathrm{mod}\ q)$,

$y = y_0^{H(CI_s, r_c)} \# y_{CA}^{H(CI_s, r_c)} \# r_c (\mathrm{mod}\ p)$

签名: 使用 SCS 签名密钥, S 计算出他对消息 m 的自证明签名 R= (r, s), 方法如下: S 随机取$k \in_R Z_R^*$, 并计算:

$r = g^k (\mathrm{mod}\ p)$, $s = x \# H(m + CI_s + r_c, r) + k (\mathrm{mod}\ q)$

则$\{(r, s), CI_s, r_c\}$就是验证人收到的 S 关于消息 m 的签名.

#验证: 验证人通过以下三步验证自证明签名$\{(r, s), CI_s, r_c\}$的有效性:

(1)计算出签名人的 SCS 公钥,

$y = y_0^{H(CI_s, r_c)} \# y_{CA}^{H(CI_s, r_c)} \# r_c (\mathrm{mod}\ p)$;

(2)用上一步计算出的 y 验证签名(r, s),

$g^s \stackrel{?}{=} y^{H(m + CI_s + r_c, r)} \# r (\mathrm{mod}\ p)$;

(3)检查签名人的长期公钥 y_0 是否与 CI_s 中的一致.

5 安全性分析

改进方案的安全性分析可以从签名人的 SCS 签名公钥入手. 原方案中的签名公钥是 $y = y_0 \cdot y_{CA}^{H(\sigma_s, r_c)} \cdot r_c \pmod p$, 式子中的长期公钥 y_0 可以被轻易地剥离掉, 比如使用 $r_c = y_0^{-1} \pmod p$. 改进方案中, 签名人的 SCS 签名公钥是 $y = (y_0 \cdot y_{CA})^{H(\sigma_s \| r_c)} \cdot r_c \bmod p$, 当 CA 试图通过构造 r_c 来伪造一个有效的自证明签名 SCS 时, 它面临的困难和试图伪造一个有效的 Schnorr 签名是一样的. 这是因为 $(y_0 \cdot y_{CA})^{H(\sigma_s, r_c)} \cdot r_c \bmod p$ 具有和 Schnorr 签名方案相同的结构, 而 Schnorr 签名在随机问答器模型下已经被证明是安全的. 所以说, 我们的改进方案可以抵抗 CA 的伪造攻击. 性能方面, 改进方案中签名人计算自证明签名私钥时比原方案增加了一次乘法运算, 验证人的计算量完全一致.

6 结论

自证明签名方案可以在不影响现有 PKI 分层结构的基础上减轻验证方的负担; 尽管签名人的负担有所加重, 但加重的部分完全可以通过预处理的办法一次性消除. 总的来说, 自证明签名所取得的优势是明显的, 并将在 PKI 中获得应用. 本文对原方案构造了一种伪造攻击, 与常见的恶意攻击者不同的是该攻击由可信任第三方发起. 这看起来有点矛盾, 但是从安全和技术的角度出发, 我们还是要尽量防止这种的存在. 本文的改进方案做到了这一点.

参考文献:

[1] C Adams, S Llyod. Understanding Public Key Infrastructure[M]. New Riders Publishing, 1999.

[2] M Myers, R Ankney, A Malpani, S Galperin, C Adams. RFC 2560, X.509 Internet public key infrastructure on2line certificate status protocol2OCSP[S]. June, 1999.

[3] B Lee, K Kim. Self2certified signature[A]. Progress in Cryptology: IN2DOCRYPT 2002[C]. LNCS 2551, Springer2Verlag, 2002. 199- 214.

[4] C P Schnorr. Efficient signatures generation by smart card[J]. Journal of Cryptology, 1991, 4(3): 161- 174.

[5] D Pointcheval, J Stern. Security arguments for digital signatures and blind signatures[J]. Journal of Cryptology, 2000, 13(3): 361- 396.

作者简介:

李新国 男, 1976 年 1 月生于河南洛阳, 现为西安电子科技大学通信工程学院密码学专业博士研究生, 主要研究方向是: 数字签名, PKI, 安全组通信等.

葛建华 男, 1961 年 9 月生于江苏南通, 现为西安电子科技大学教授, 博士生导师, 主要研究方向包括: 数字电视, 通信系统, 信号处理, 密码学等.