签密的仲裁安全与仲裁安全的签密方案

粟 栗,崔国华,李 俊,郑明辉
(华中科技大学计算机学院,湖北武汉 430074)

摘 要: 签密能高效地同时完成数据加密与认证,可用于设计紧凑的安全通信协议.签密中的仲裁机制用于保护签密的不可抵赖性,但同时用于仲裁的信息可能危及协议安全.本文指出签密仲裁中存在仲裁者解密攻击和仲裁机制无法保护明文完整性两种安全隐患,归纳其原因并指出解决方法.提出一个可安全仲裁的安全混合签密方案 *SASC*,并在随机预言机模型下证明 *SASC* 方案具有 IND-CCA2 和 UF-CMA 安全性;*SASC* 基于明文仲裁,不仅能维护明完整性而且能抵抗仲裁者解密攻击. *SASC* 方案不增加计算量和通信量,且对明文的长度没有限制.

关键词: 签密; 仲裁安全; 随机预言模型; 可证明安全性
中图分类号: TP309 **文献标识码:** A **文章编号:** 0372-2112 (2007) 11-2117-06

Arbitral Security of Signcryptions and a Securely Arbitral Signcryption Scheme

SU Li, CUI Guo-hua, LI Jun, ZHENG Ming-hui
(*Department of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, Hubei* 430074, *China*)

Abstract: Signcryption provides confidentiality and authenticity efficiently; it can be used to design compact communication protocol. Arbitration mechanism is used for settling disputes in signcryption, but the information that the judge gets also brings some security problems. This paper points out two problems: in some scheme, the arbitrator can decrypt all the signcryptions of a receiver while he gets some kinds of arbitration message; in another schemes, the arbitration mechanism cannot protect the integrity of plaintext. Analyze the two kinds of problems and concludes their reasons separately, we proposed a resolvent that can solve the two problem by changing a secure arbitration message. Based on the attack and analysis, this paper proposes a secure arbitral signcryption (*SASC*) scheme and proves its IND-CCA2 security and UF-CMA security in random oracle model. Furthermore, *SASC* is a securely arbitral signcryption scheme, it can protect the integrity of plaintexts by an arbitration message associated with plaintext; and the scheme can resist decryption attacks of arbitrator, even he gets the arbitration message. *SASC* does not increase computation nor communication overloads; it has no limitation to the length of plaintext, which makes *SASC* more convenient. Proofs and analysis show that *SASC* is an efficient and secure scheme.

Key words: signcryption, security of arbitration, random Oracle, provable security

1 引言

签密[1]是同时实现信息加密和认证的新密码技术,其计算量和通信量低于加密并签名.基于签密设计的通信协议不仅计算量小,且能在无可信中心参与的情况下保护数据的不可伪造性和非否认性[2],在电子邮件[3]、分布式密钥分配[4]、动态无线局域网等方面有广泛的应用价值.

签密是一个三方协议,包括发送者 Alice、接收者 Bob 以及一个用于解决纠纷的仲裁者 Charlie.签密的基本安全性包括[1]:(1)不可伪造性:包括 Bob 在内的任何一个攻击者都不能伪造一个 Alice 的有效签密;(2)机密性:通过 Alice 发送给 Bob 的签密不能计算 Alice 或 Bob 的任何私有信息;(3)不可抵赖性:当 Alice 否认发送一个密文给 Bob 时,Charlie 能在 Bob 的帮助下通过有效计算证实这个签密的发送方和接收方.当出现 Alice 否认签密的情况时,Bob 可向 Charlie 申请仲裁.此时,Bob 需要发送一些用于仲裁的必要信息给 Charlie,但仲裁信息可能影响签密协议的安全性.

本文分析文献[1,5~8]中的仲裁信息,指出两类安全隐患:(1)文献[1,5,6]方案中,存在仲裁者解密攻击,若仲裁 Alice 发送给 Bob 的一个有效签密,Charlie 能解密所有 Alice 发送者 Bob 的有效签密;(2)文献[7,8]方案中的密文验证机制则不能保护明文的完整性.对于两

收稿日期:2006-11-15;修回日期:2007-06-25
基金项目:国家自然科学基金(No.6040,3027),湖北省教育厅优秀中青年项目(No.Q200629001)

种安全隐患,本文分别进行分析,并指出安全的签密仲裁应能维护明文完整性并抵抗仲裁者解密攻击.提出可安全仲裁的签密模型 *SASC*(Secure Arbitral SignCryption);在随机预言(Random Oracle)模型下证明 *SASC* 方案满足选择密文攻击下的不可区分性(IND-CCA2)和选择消息攻击下的不可伪造性(UF-CMA)[5];证明 *SASC* 具有仲裁安全性,且仲裁能保护明文的完整性.*SASC* 方案没有增加计算量和通信次数,且对明文的长度没有限制,是一个高效且安全的签密方案.

2 签密中的仲裁安全性分析

2.1 参数设置

k 为安全参数,p 是满足安全参数的大素数,即 $|p|=k$;选择一个大素数 q 满足 $q|(p-1)$ 并且 $|q|>l_q$,其中 l_q 子群安全模数 $|q|$ 的比特长度,g 是 Z_p^* 中的 q 阶生成元;

Alice 为发送者,拥有私钥 $x_A\in Z_q^*$ 和公钥 $y_A=g^{x_A} \bmod p$;

Bob 是接收者,拥有私钥 $x_B\in Z_q^*$ 和公钥 $y_B=g^{x_B} \bmod p$;

$SYM=\{E(\cdot),D(\cdot)\}$ 是 IND-CPA 安全的对称加密算法,其中 E 是加密函数,D 是解密函数;

$SIG=\{SIG.gen(\cdot),SIG.ver(\cdot)\}$ 是 UF-CMA 安全的三元组签名算法,$SIG.gen(\cdot)$ 和 $SIG.ver(\cdot)$ 分别为签名产生算法和签名验证算法;

$G(\cdot)$ 和 $H(\cdot)$ 是两个随机预言机,在算法的具体实现中以安全的 hash 函数代替.

2.2 仲裁者解密攻击及分析

对文献[1,5,6]中的方案,若 Charlie 进行一次有效仲裁后,便能解密 Alice 发送给 Bob 的所有签密[10],称为仲裁者解密攻击.以改进的 *SDSS* 方案[5](*MSDSS*)为例进行攻击和分析如下.

$MSDSS.sc_{x_A,y_B}^{G,H}(m)$

$bind\leftarrow y_A\|y_B$,
$x\leftarrow_R Z_q^*$, $K\leftarrow y_B^x \bmod p$,
$\tau\leftarrow G(K)$, $c\leftarrow E_\tau(m)$,
$r\leftarrow H(m\|bind\|K)$,
$s\leftarrow x/(r+x_A) \bmod q$
$C\leftarrow(c,r,s)$
return C

$MSDSS.usc_{y_A,x_B}^{G,H}(C,y_A)$

分解 C 为 (c,r,s)
if ((c,r,s)不在正确的空间) return "Reject"
else
  $w=(y_Ag^r)^s \bmod p$; $K\leftarrow w^{x_B} \bmod p$;
  $\tau\leftarrow G(K)$; $m\leftarrow D_\tau(c)$, $bind\leftarrow y_A\|y_B$
if ($H(m\|bind\|K)=r$ return m
else return "Reject"

MSDSS 中,若一个签密 $C=(c,r,s)$ 需要仲裁,Bob 发送 $K=y_B^x \bmod p$ 给 Charlie,Charlie 能求解密密钥 τ,并对 C 进行解密和验证.由于 Charlie 因仲裁得到 K,他可用 y_B 为底,重构验证等式得 $y_B^x=(y_B^{x_A},y_B^r)^s$.因为 $K=y_B^x \bmod p$,由等式可得 $K^{s^{-1}}\cdot y_B^{-r}=y_B^{x_A}=g^{x_Ax_B}$.对 C 执行仲裁后,Charlie 可对于 Alice 发送给 Bob 的任何一个有效签密 $C^*=(c^*,r^*,s^*)$ 解密攻击如下:

(1)以 y_B 为底重构 $y_B^{x^*}=(y_B^{x_A}\cdot y_B^{r^*})^{s^*}$ 成立,且 r^*,s^*,$g^{x_Ax_B}$ 已知,故可求得 $K^*=y_B^{x^*} \bmod p$;

(2)计算 $\tau^*\leftarrow G(K^*)$;

(3)解密 $m^*\leftarrow D_{\tau^*}(c)$.

MSDSS 虽然在 FUO(Flexible Unsigncryption Model)模型下证明具有具有 IND-CCA2 安全性和 UF-CMA 安全性,但仍不能抵抗 Charlie 在仲裁后执行解密攻击.

记三元组签名为 (r,e,s),其中 $r=g^l \bmod p$,l 为随机数;e 为散列值.由安全构造法则[11]可知:s 是以 (e,x_A,l) 为参数的线性函数值,且 l 和 x_A 不能相乘.对任意的三元组签名算法,由 $((g,y_A),(e,s))$ 和验证等式能恢复 r,且 (e,s) 都是指数项.故能通过 $((g^{x_B} \bmod p, y_A^{x_B} \bmod p),(e,s))$ 和验证等式能恢复 $r^{x_B} \bmod p$.由于 $g^{x_B} \bmod p=y_B$ 可公开获取,若仲裁者获得 $g^{x_Ax_B}$,则必能以 y_B 为底重构验证等式,恢复 $r^{x_B} \bmod p$,进而计算解密密钥.分析并归纳仲裁者解密攻击的必要条件如下.

结论 1 (仲裁者解密攻击的必要条件):仲裁者解密攻击需要两个的必要条件:

(1)仲裁者通过仲裁签密能获得 $g^{x_Ax_B}$;

(2)对任意一个有效签密,仲裁者能恢复其中包含的承诺值 r.

2.3 明文完整性维护

在文献[7,8]中提出了可公开验证的签密方案,签密者先加密明文生成 c,然后对密文 c 签名生成 σ,(c,σ) 作为签密发送给接收者.基于签名的公开可验证性,并适当配置签名中的信息,任何人都可以验证签密的有效性并判断发送者和接收者.

文献[7,8]中的签密算法仅相当于对密文进行签名,由于在签密验证和仲裁过程中不涉及明文内容,不能保护明文完整性,签密收发双方必将在明文内容上出现分歧.譬如在电子支票协议中,发送者可以任意宣称支票金额,而验证或仲裁都无法核查其明文.密文可公开验证的签密不能保证明文完整性,签密者可能发送错误信息或传播非法信息,给网络中的信息安全带来隐患.

综上可得,一个安全的签密不仅需要满足内部安

全性(FUO-IND-CCA2 和 UF-CMA 安全性),还应能抵抗仲裁者攻击,并通过仲裁机制保护明文完整性.

3　安全性定义与困难问题假设

定义签密方案 $Signcryption = \{SC(\cdot), USC(\cdot)\}$, $SC(\cdot)$和 $USC(\cdot)$分别表示签密和解签密预言机.

定义 1　(签密的 FUO-IND-CCA2 安全性):假定攻击者为 A_c,由 find 和 guess 两个阶段组成,分别用 A_1 和 A_2 表示. A_c 可以有限次询问随机预言机、*SC* 预言机和 *USC* 预言机.若在 A_1 选定的两个明文(m_0, m_1)中任选一个进行签密生成 C,而 A_2 无法在多项式时间内以不可忽略的概率分辨 C 是对哪一个明文的签密,则称签密方案满足 FUO-IND-CCA2 安全性.

定义 2　(签密的 UF-CMA 安全性):定义攻击者为 A_{UF}. A_{UF}能任意选择明文并有限次询问随机预言机和 *SC* 预言机.若在多项式时间内,攻击者 A_{UF}不能以不可忽略的概率成功伪造一个未询问 *SC* 的明文 m 的签密,则称签密方案满足 UF-CMA 安全性.

定义 3　(对称加密函数的 IND-CPA 安全性):对称加密函数为 $SYM = (E(\cdot), D(\cdot))$, E 和 D 为加密和解密函数.攻击者 A_p 由 find 和 guess 两个阶段组成,分别用 A_{p1}和 A_{p2}表示. A_{p1}能有限次询问 E 并输出两个明文(m_0, m_1),若从中任选一个加密生成密文 c,而 A_{p2}无法在多项式时间内以不可忽略的概率分辨密文 c 是对哪一个明文的加密,则称 *SYM* 满足 IND-CPA 安全性.

定义 4　(签名的 UF-CMA 安全性):假定对签名方案为 $SIG = \{SIG.gen(\cdot), SIGN.ver(\cdot)\}$ 的攻击者为 A_{SIG},他能有限次任意选择明文并询问签名预言机 $SIG.gen(\cdot)$.若在多项式时间内,攻击者 A_{SIG}不能以不可忽略的概率成功伪造一个明文 m 的签名通过 $SIG.ver(\cdot)$验证,且 m 未询问过签名预言机,则称签密方案满足 UF-CMA 安全性.

定义 5　(GDH 困难问题假设):假定 A_{gdh}为 *GDH* 问题攻击者, A_{gdh}的目的是在 *DDH* 预言机的帮助下解决 *CDH* 问题.即已知 $X = g^x$ 和 $Y = g^y$,在多项式时间内求解目标 DH 值 g^{xy}. A_{gdh}可以询问 *DDH* 预言机 q_{ddh}次,若询问的项(X, Y, Z)是一个 *DDH* 组,则预言机返回 1,否则返回 0.设 A_{gdh}在多项式时间 t 内破解 GDH 问题的优势为 $Adv^{invert}_{GDH}(k, t, q_{ddh})$,可知 Adv^{invert}_{GDH}是可忽略的概率.

4　仲裁安全的签密方案 SASC

由三元组签名的性质,对于 m 的三元组签名(r, e, s),以(r, s)或(e, s)作为签名具有同等效力[11].通过 2.2节的分析可知:若要抵抗仲裁者解密攻击,至少需要破坏攻击成立的两个必要条件之一.

(1)若(e, s)为 m 的签名,则需要使仲裁者不能重构验证等式恢复 r.基于三元组签名的基本准则,达到此目的需要改变签名算法或增加额外的幂运算来保护 r.

(2)若(r, s)为 m 的签名,我们可以在 e 的生成过程中增加秘密信息.由于签密过程中已存在 DH 秘密 y_B^x,可以方便的利用来构造安全的 e 值,使得除 Bob 外的任何人都不能计算 e,从而不能重构签名验证等式.采用以(r, s)为签名可不需更改签名算法和增加计算量,便能避免仲裁者攻击.

选取已证明 UF-CMA 安全的 Schnorr 签名算法,并修改其中的 e 值构造,设计仲裁安全的签密方案 *SASC* (Securely Arbitral Signcryption)如下.

$SASC.SC^{G,H}_{x_A,x_B}(m)$

$x \leftarrow_R Z_q^*, r = g^x \bmod p; \kappa \leftarrow y_B^x \bmod p,$
$K \leftarrow G(y_A \| \kappa); c \leftarrow E_K(m),$
$e \leftarrow H(r \| m \| \kappa); s \leftarrow x - x_A e \bmod q$
$C \leftarrow (c, r, s)$
return C

$SASC.USC^{G,H}_{y_A,x_B}(C, y_A)$

分解 C 为(c, r, s)并检查所有是否在定义域
if ((c, r, s)不在正确的定义域)return "Reject"
else
$\kappa \leftarrow r^{x_B} \bmod p; K \leftarrow G(y_A \| \kappa); m \leftarrow D_K(c)$
if $(e = H(r \| m \| \kappa)$且 $g^s = r \cdot y_A^e \bmod p)$ return m
else return "Reject"

若 Alice 否认一个明文,Bob 发送签密 C 和 $\kappa = y_B^x \bmod p$ 给 Charlie 进行仲裁. Charlie 利用 κ 计算 K,解密 c 得 m,然后并验证 $e = H(r \| m \| \kappa)$是否成立.若验证成立,则支持 Bob 的申诉,且 Alice 不能否认签密 C 是对 m 的签密.

与目前已有的其它安全的混合签密方案比较相比,*SASC* 方案具有如下特点:

(1)不仅满足 FUO-IND-CCA2 安全性和 UF-CMA 安全性,并能保证仲裁安全抵抗仲裁者解密攻击和维护明文的完整性;

(2) *SASC* 方案的签密、解签密的计算量之和为 5 次模幂运算,其中签密需 2 次模幂运算,解签密需 3 次.与 MSDSS[5]、DHEtS[7], New-DSA[9] 等可证明安全的签密方案计算量相同,但安全性更高;

(3) *SASC* 方案中的仲裁仅需要一次 Hash 运算和一次对称解密,有利于仲裁机构的迅速仲裁;

(4) *SASC* 方案的通信次数仅为两次;

(5) *SASC* 方案中的加密和签名算法都可分别用任意 IND-CPA 安全的对称加密算法和 UF-CMA 安全的签名算法替换,具有良好的适应性和高度的灵活性;

(6) *SASC* 方案对明文 m 的长度没有限制,可以用于大数据的签密,且计算量和附加通信量不随明文的

增加而增长.

5 安全性证明

5.1 预言机定义和分析

G 和 H 预言机是对随机预言机的仿真.对 G 预言机有两种不同的询问,并分别记录于 L_1^G 和 L_2^G.(1)hash 询问.输入询问为任意长度的字符串 λ_i,输出一个随机数 K_i 为询问结果,将(λ_i, K_i)存入 L_1^G;(2)签密预言机中的 G 询问.签密预言机不拥有 x_B,也不能计算 $r_i^{x_B}$,签密预言机得到(r_i, y_A)后,G 预言机给出一个随机值 K_i 代替 $G(y_A \| r_i^{x_B})$.用"?"表示未知的 $r_i^{x_B}$,并将$(r_i, y_A, ?, K_i)$存入 L_2^G.

H 预言机也有两种查询和记录.(1)hash 询问.输入任意长度的字符串 μ_i,随机输出 e_i 为 $H(\mu_i)$,并将(μ_i, e_i)存在于 L_1^H;(2)由加密预言机和解密预言机中的 H 询问.输入为 r_i,输出随机数 e_i 为 $H(r_i \| m_i \| r_i^{x_B})$值,以"?"代替未知的 $r_i^{x_B}$,记 $u_i = (r_i \| m_i \| ?)$,存储(r_i, μ_i, e_i)于 L_2^H.

G-sim(L^G, λ)
分解 λ 为 $y \| \kappa$,κ 是 λ 中右 k 位;
if $(DDH_g(X, y_B, \kappa) = 1)$ return NULL
else if (L_2^G 中存在$(r_i, y_i, ?, K_i)$满足 $DDH_g(r_i, y_B, \kappa) = 1$ 且 $y = y_i$)
return K_i
else if (L_1^G 中存在(λ_i, K_i)使 $\lambda = \lambda_i$)
return K_i
else $K_i \leftarrow_R \{0,1\}^l$
return K_i;
$\kappa_i = \kappa$;将(λ_i, K_i)加入 L_1^G

H-sim(L^H, u)
分解 μ 为 $r \| m \| \kappa$,κ 是 μ 中右 k 位
if $(DDH_g(x, y_B, \kappa) = 1)$ return NULL
else if (L_2^H 中存在(r_i, u_i, e_i)满足 $DDH_g(r_i, y_B, \kappa) = 1$ 且 $r_i \| m_i = r \| m$)
return e_i
else if (L_1^H 中存在(μ_i, e_i)满足 $\mu = \mu_i$)
return e_i
else $e_i \leftarrow_R Z_q^*$
return e_i;
$\mu_i = \mu$;将(μ_i, e_i)加入 L_1^H

通过 G 和 H 预言机,可以仿真签密和解签密预言机.当对信息 m 提出的签密通过签密预言机 *SC-sim* 实现;对签密 C 提出解签密请求时,利用解签密预言机 *USC-sim* 模拟实现.

SC-sim$(L_2^G, L_2^H, y_A, y_B, m)$
$K \leftarrow_R \{0,1\}^l$; $c \leftarrow E_{\langle K \rangle}(m)$;
$e \leftarrow_R Z_q^*$; $s \leftarrow_R Z_q^*$; $r \leftarrow g^s y_A^e$
$r_i \leftarrow r$; $K_i \leftarrow K$; $m_i \leftarrow m$; $e_i \leftarrow e$;
将$(r_i \| m_i \| ?, e_i)$加入 L_2^H
$C \leftarrow (c, r, s)$
return C

USC-sim$(L^G, L^H, X, \bar{y}_A, y_B, C)$
分解 C 为(c, r, s);
if $r = X$ return NULL;
if ($\exists (\lambda_i, K_i) \in L_1^G$ 满足$(\lambda_i = \bar{y}_A \| \kappa_i, DDH_g(r, y_B, \kappa_i) = 1)$ or
$(\exists (r_i, y_i, ?, K_i)) \in L_2^G$ 满足$(r = r_i, y_i = \bar{y}_A)$) then $K = K_i$;
else $K \leftarrow_R \{0,1\}^l$;
$r_i \leftarrow r$; $K_i \leftarrow K$;将$(r_i, y_i, ?, K_i)$加入 L_2^G; $m \leftarrow D_{\langle K \rangle}(c)$;
if ($\exists (\mu_i, e_i) \in L_1^H$ 满足 $DDH_g(r, y_B, \kappa_i) = 1, \kappa_i$ 为 μ_i 右 k 位) or
$(\exists (r_i, u_i, e_i) \in L_2^H$满足$(r = r_i, m = m_i))$ then $e' = e_i$;
else $r_i \leftarrow r$, $m_i \leftarrow m$
置 $e' \leftarrow_R Z_q^*$, $e_i = e'$,并将$(r_i \| m_i \| ?, e_i)$加入 L_2^H
if $SIG.ver(r, e, s, y_A) = 1$ then return m
else return NULL

仿真环境与真实环境可能在一定情况下有差异,此时仿真失败.定义 $q_g, q_h, q_{sc}, q_{usc}$ 分别为查询问 G 预言机,H 预言机,加密预言机和解签密预言机的次数;所花费的时间分别为 $t_g, t_h, t_{sc}, t_{usc}$.记仿真失败的事件为 *Bad*,发生 *Bad* 的情况可以分为两类:

(1)当预言机向 $DDH_g(\cdot,\cdot,\cdot)$询问目标 DH 值 $\kappa = g^{xy}$时,返回 NULL,攻击者因此能判断所询问的值为目标 DH 值,从而 GDH 问题解决,称为 *GDHBrk* 事件,记其概率为 $Succ_{GDH, A_{gdh}}^{invert}(k)$;

(2)由于 *SC-sim* 和 *USC-sim* 中需要查询 $\kappa = r^{x_B} \bmod p$ 的散列值,而由于模拟器不知道 x_B 不能正确计算 κ 而以"?"表示,并列为新条目.而 κ 值可能已经查询过 H 或 G,或正好为目标 *DH* 值.记为 $Bad \wedge \neg GDHBrk$,分解 $Bad \wedge \neg GDHBrk$ 为 $SCBad \vee USCBad$.

(a) *SCBad*:若 L_2^G 中的"?"可能是 G 或 H 预言机中已查询的一个条目,也可能正好为 DH 值,因此可能造成错误. *SC-sim* 出现 *Bad* 的概率:$\Pr[SCBad] \leqslant q_{sc}(q_g + q_h + 1)/2^{l_q(k)}$.

(b) *USCBad*:不重复考虑 *SCBad*,仅需考虑 *uscbAD* $\wedge \neg SCBad$.在解签密过程中,当随机选择 e' 且正好 $e = e'$ 时,该签密有效且不是由 *SC-sim* 签发的.因此 $USCBad \wedge \neg SCBad$ 事件出现的概率小于伪造成功的概率:$\Pr[USCBad \wedge \neg SCBad] \leqslant Adv_{SC}^{UF\text{-}CMA}(\cdot)$.

综合两类情况,可知仿真环境中 *Bad* 事件发生的可能性是可忽略的,且:

$$\Pr[Bad] \leqslant Succ_{GDH, A_{gdh}}^{invert}(k) + q_{sc}(q_g + q_h + 1)/2^{l_q(k)} + Adv_{SC}^{UF\text{-}CMA}(\cdot) \quad (1)$$

5.2 *SASC* 的 FUO-UF-CMA 安全性

假定 *SASC* 的伪造攻击者 A_{UF}能在多项式时间 t 内以不可忽略的概率 $Adv_{SC}^{UF\text{-}CMA}(\cdot)$伪造一个有效签密,则利用 A_{UF}的能力,可构造一个签名伪造攻击者 A_{SIG}以不可忽略的概率伪造一个有效签名.在 FUO-UF-CMA 的攻击中,A_{UF}不能询问解签密预言机.

Adversary $A_{SIG}(k,p,q,g,y_A,x_B,y_B)$

$e^* \leftarrow_R Z_q^*; s^* \leftarrow_R Z_q^*; x_B \leftarrow_R Z_q^*; y_B \leftarrow g^{x_B}, y_A^* \leftarrow (r^* \cdot g^{-s^*})^{(e^*)^{-1}}$

Run $A_{UF}(k,find,y_A,y_B,x_B)$,用 *G-sim*, *H-sim* 和 *SC-sim* 回答 A_{UF}的询问

$A_{UF}(k,find,y_A,y_B,x_B)$输出$(m,C)$

if *USC-sim*$(C)=m$ 且 m 从未询问过 *SC-sim*　return 1

else return 0

通过签密 $C=(c,r,s)$中的 r 值是否询问过 *G-sim* 或 *H-sim* 进行分类,若 A_{UF}输出的为有效签密,则依据 r 可划分为:$r\in\neg Ask_G \vee Ask_G=\neg Ask_G \vee (Ask_G \wedge \neg Ask_H) \vee (Ask_G \wedge Ask_H)$.

(1)$r\in\neg Ask_G$. 由于 r 未询问过 G 预言机,对 C 进行解签密及验证时不能从 L^G 中获取 K 值. G 预言机随机选择一个解密密钥 K 并解密得到 m'. 由于 *SYM* 是 IND-CPA 安全的,可知由 K 解密 c 得出的 m' 也是随机的,此时 $m'=m$ 的概率为:$\Pr[A_{UF}wins \wedge \neg Ask_G]=1/2^{l_q}$.

(2)$r\in Ask_G \wedge \neg Ask_H$. r 是 *SC* 预言机对另一个明文签密产生的承诺,并已经给出了随机值 K 作为密钥,解密得到的 m' 必然等于 m. 但若未查询 *H-sim*,则 *H-sim* 随机选择一个 e,并验证签名,此时随机选择一个 e 并使签名验证成功的概率为:$\Pr[A_{UF}wins \wedge Ask_G \neg Ask_H]=1/2^{l_q}$.

(3)$r\in Ask_G \wedge Ask_H$. 此时,攻击者可获取 K,m,r,e. 由于 $Ask_H=1$,说明已有 $e=\mathrm{H}(r \| m \| r^{x_B})$询问过 H 预言机,且由于攻击者不能访问解签密预言机,故攻击者必然已经获取 r^{x_B}并向 H 预言机提出过关于 $H(r \| m \| r^{x_B})$的询问,获得 e;且攻击者还能成功构建一个 s 满足签密验证等式.

在构造有效签密的同时,A_{UF}同时构造了一个有效的签名(m,e,s)且未查询过签名预言机 *SIG*. *gen*,攻击者伪造签名成功. 由签名定义可知,其概率不大于 $Adv_{SIG}^{UF\text{-}CDA}(k,t,q_g,q_h)$.

综上可得:$Adv_{SASC}^{UF\text{-}CDA}(k,t,q_g,q_h,q_{sc}) \leqslant Adv_{SIG}^{UF\text{-}CDA}(k,t,q_g,q_h)+1/2^{l_g-1}$

由于 *SIG* 是 UF-CMA 安全的签名方案,$Adv_{SIG}^{UF\text{-}CDA}(k,t,q_g,q_h)$是可忽略的概率;$1/2^{l_g-1}$也是可忽略的概率,故 *SASC* 方案具有 UF-CMA 安全性.

5.3 *SASC* 的 FUO-IND-CCA2 安全性

设攻击者 A_c 由 find 阶段算法 A_1 和 guess 阶段算法 A_2 组成.若 A_c 能在多项式时间 t 内以不可忽略的概率对 *SASC* 成功执行 FUO-IND-CCA2 攻击,那么利用 A_c 的优势,可构造攻击者 A_{gdh}以不可忽略的概率破解 GDH 问题.

Adversary $A_{gdh}(k,p,q,g,X,Y)$

$e^* \leftarrow_R Z_q^*; s^* \leftarrow_R Z_q^*; r^* \leftarrow X; y_B \leftarrow Y, K^* \leftarrow_R \{0,1\}^l; y_A^* \leftarrow (r^* \cdot g^{-s^*})^{(e^*)^{-1}}$

Run $A_1(k,find,y_A,y_B)$,用 *G-sim*, *H-sim*, *SC-sim* 和 *USC-sim* 回答 A_1 的询问

$A_1(k,find,y_A,y_B)$输出(m_0,m_1,s)

$b \leftarrow_R \{0,1\}; c^* \leftarrow E_{\langle K^*\rangle}(m_b); C^* \leftarrow (c^*,e^*,s^*)$

Run $A_2(k,guess,m_0,m_1,C^*,y_A,y_B,s)$用 *G-sim*, *H-sim*, *SC-sim* 和 *USC-sim* 回答 A_2 的询问 $A_2(k,guess,m_0,m_1,C^*,y_A,y_B,s)$输出 b'

return κ^*

除 *Bad* 事件外,模拟环境与理想的情况没有任何区别,若 A_c 成功且不出现 *Bad* 事件则 FUO-IND-CCA2 攻击成功.基于 *Bad* 事件可得:

$$\Pr[A_c\ wins \wedge \neg Bad] \geqslant \Pr[A_c\ wins]-\Pr[Bad] \geqslant \frac{1}{2}+\frac{1}{2}Succ_{SC,A_c}^{fuo\text{-}ind\text{-}cca2}(k)-\Pr[Bad] \quad (2)$$

同理,基于 *GDHBrk* 事件进行分解可得:

$$\Pr[A_c\ wins \wedge \neg Bad]-\Pr[A_c\ wins \wedge \neg Bad \wedge \neg GDHbrk] \leqslant Succ_{GDH,A_{gdh}}^{invert}(k) \quad (3)$$

若 A_c 成功且不出现 *Bad* 和 *GDHBrk* 事件,则可构建一个 IND-CPA 攻击者 $A_p=(A_{p1},A_{p2})$和下面的 *sim-cpa* 模拟环境,A_p 能利用 A_c 的能力破解 IND-CPA 安全的对成加密函数 *SYM*.

Adversary $A_p(k,p,q,g)$

$e^* \leftarrow_R Z_q^*; s^* \leftarrow_R Z_q^*; x \leftarrow_R Z_q^*; r^*=X \leftarrow g^x; x_B \leftarrow_R Z_q^*; y_B \leftarrow g^{x_B}; y_A^* \leftarrow (r^* \cdot g^{-s^*})^{(e^*)^{-1}}$

Adversary $A_{p1}(k,find,y_A,y_B)$

Run $A_1(k,find,y_A,y_B)$,用 *G-sim*, *H-sim*, *SC-sim* 和 *USC-sim* 回答 A_1 的询问

$A_1(k,find,y_A,y_B)$输出(m_0,m_1)

Return(m_0,m_1,r^*,s^*)

$b \leftarrow_R \{0,1\}; K^* \leftarrow_R \{0,1\}^l; c^* \leftarrow E_{\langle K^*\rangle}(m_b)$

Adversary $A_{p2}(k,guess,m_0,m_1,c^*,r^*,s^*)$

Run $A_{p2}(k,guess,m_0,m_1,c^*,r^*,s^*,y_A,y_B)$,用 *G-sim*, *H-sim*, *SC-sim*, *USC-sim* 回答 A_2 的询问

$A_2(k,guess,m_0,m_1,C^*,y_A,y_B)$输出 b'

return b'

对 A_c 来说,在$\neg Bad \wedge \neg GDHbrk$ 的情况下,*sim-cpa* 环境与 *sim* 环境是不可区分的.因此如果 $A_c wins \wedge \neg Bad \wedge \neg GDHbrk$ 发生则事件 $A_p wins$ 发生.因此有:

$$\Pr[A_c wins \wedge \neg Bad \wedge \neg GDHbrk]_{sim} \leqslant \frac{1}{2}+\frac{1}{2}Succ_{SYM}^{ind\text{-}cpa}(l,t_2,0) \quad (4)$$

由等式(1)~(4)合并可得:

$$Adv_{SASCC,A_c}^{fuo\text{-}ind\text{-}cca2}(k,t,q_g,q_h,q_{sc},q_{usc}) \leqslant 4Adv_{GDH}^{invert}(k,t_1,q_{ddh})+Adv_{SYM}^{IND\text{-}CPA}(l,t_2)+2Adv_{SC}^{UF\text{-}CMA}(\cdot)+\frac{q_{sc}(q_g+q_h+1)}{2^{l_q(k)-1}}$$

由 GDH 问题是困难的,*SIG* 是 UF-CMA 安全的签名方案,SYM 具有 IND-CPA 安全性,且 $q_{sc}(q_g+q_h+1)/2^{l_q(k)-1}$是可忽略的概率,因此 $Adv_{SASC}^{FUO\text{-}IND\text{-}CCA2}(\cdot)$的概率

是可忽略的,即 *SASC* 方案满足 FUO-IND-CCA2 安全性.

5.4 仲裁安全性

SASC 中的仲裁能保护明文完整性. *SASC* 方案的仲裁中,接收方 Bob 通过发送 κ 给 Charlie.通过计算,Charlie 可以获得(m,r,s,κ),并通过签名验证等式 $r=g^s\cdot y_A^{H(m,r,k)} \bmod p$ 验证签密的有效性.因此 *SASC* 能保护签密中明文的完整性,发送者不能欺骗接收者或任意发送违规的信息.

SASC 可以抵抗仲裁者攻击.即使因为仲裁签密(c,r,s),Charlie 获得 κ,但 Charlie 不能对 Alice 发送给 Bob 的其它签密进行解密攻击.即使 Charlie 仲裁了一个有效签密 C,对于另一个 Alice 发送给 Bob 的签密 $C^*=(c^*,r^*,s^*)$,若要重构验证等式获得 κ^* 必须已知 e^*;但若攻击者不知道 κ^* 则无法对对 G 预言机提出正确的 hash 询问 $G(y_A\|\kappa^*)$获得 e^*,形成矛盾. *SASC* 中通过类似"死锁"的机制使仲裁者不能重构验证等式,因而无法执行解密攻击.

6 总结

签密的仲裁机制用于解决纠纷,维护签密的不可抵赖性;仲裁所需的信息可能危及签密协议的安全.本文分析签密仲裁信息可能导致仲裁者解密攻击和不能保证明文完整性两个安全隐患,并给出其原因.指出应通过构造安全的散列值避免仲裁者攻击;通过明文仲裁约束明文完整性.设计了可安全仲裁的方案 *SASC*,并在随机预言模型下证明 *SASC* 具有 FUO-IND-CCA2 安全性和 UF-CMA 安全性;且 *SASC* 基于明文仲裁,维护了明文完整性和不可否认性,并能抵抗仲裁者解密攻击.与现有的混合签密方案相比,*SASC* 的计算量和通信次数没有增加,且具有更高的安全性,高度的灵活性,可方便地用于设计安全而紧凑的协议.

参考文献:

[1] Zheng Y L. Digital signcryption or how to achieve cost(signature & encryption)〈cost(siganture) + cost(encryption)[A]. Kaliski B D. Advances in Cryptology-CRYPT0'97[C]. Berlin: Springer-Verlag, 1997. LNCS 1294: 165 – 179.

[2] Zheng Y L, Imai H. Using signcryption to build compact and unforgeable key establishment over an ATM network[A]. Proceedings of IEEE INFOCOM'98[C]. San Francisco: CA, 1998. 411 – 418.

[3] 王彩芬,贾爱库,刘军龙.基于签密的多方认证邮件协议[J].电子学报,2005,33(11):2070 – 2073.
Wang Caifen, Jia Aiku, Liu Junlong. Multi-party certified mail protocol based on signcryption[J]. Acta Electronica Sinica, 2005, 33(11): 2070 – 2073. (in Chinese)

[4] 陈伟东,冯登国.签密方案在分布式协议中的应用[J].计算机学报,2005,28(9):1421 – 1430.
Chen Weidong, Feng Dengguo. Some applications of signcryption schemes to distributed protocols[J]. Chinese Journal of Computers, 2005, 28(9): 1421 – 1430. (in Chinese)

[5] Baek J, Steinfeld R, Zheng Y L. Formal Proofs for the Security of Signcryption[A]. Naccache D. PKC 2002[C]. Berlin: Springer-Verlag, 2002. LNCS 2274: 80 – 98.

[6] Lee M K, Kim D K, Park K. An authenticated encryption scheme with public verifiability[A]. 5th Japan-Korea Joint Workshop on Algorithms and Computation[C]. Tokyo: IEEE Press, 2000. 49 – 56.

[7] Jeong I R, Jeong H Y, Rhee H S. Provably secure encrypt-then-sign composition in hybrid signcryption[A]. Lee P J and Lim C H. ICISC 2002[C]. Berlin: Springer-Verlag, 2003. LNCS 2587: 16 – 34.

[8] 张串绒,肖国镇.一个可公开验证签密方案的密码分析和改进[J].电子学报,2006,34(1):177 – 179.
Zhang Chuanrong, Xiao Guozhen. Cryptanalysis and improvemen t of a signcryption scheme with public verifiability[J]. Acta Electronica Sinica, 2006, 34(1): 177 – 179. (in Chinese)

[9] Shin J B, Lee K, Shim K. New DSA-verifiable signcryption schemes[A]. Lee P J and Lim C H. ICISC 2002[C]. Berlin: Springer-Verlag, 2003. LNCS 2587: 35 – 47.

[10] Petersen H, Michels M. Cryptanalysis and improvement of signcryption schemes[J]. IEE Computers and Digital Communications, 1998, 145(4): 149 – 151.

[11] Harn L, Xu Y. Design of generalized Elgamal type digital signature schemes based on discrete logarithm[J]. Electronics Letters, 1994, 30(24): 2025 – 2026.

作者简介:

粟 栗 男,1981 年生于湖北省公安县,华中科技大学计算机学院博士生.主要研究方向为公钥密码、安全协议及可证明安全性.
E-mail: hustsuli@gmail.com

崔国华 男,1947 年生于江苏太仓,教授、博士生导师,现任教于华中科技大学计算机学院.主要研究方向为密码学、信息安全.

李 俊 男,1979 年生于湖北省武汉市,华中科技大学计算机学院博士生.主要研究方向为公钥密码,秘密分享.

郑明辉 男,1972 年生于湖北省嘉鱼县,华中科技大学计算机学院博士生,湖北民族学院副教授.主要研究方向为公钥密码、密钥管理.