Journal of Computer Applications
计算机应用,2013,33(12):3514-3518
ISSN 1001-9081
CODEN JYIIDU
2013-12-01
http://www.joca.cn

文章编号:1001-9081(2013)12-3514-05
doi:10.11772/j.issn.1001-9081.2013.12.3514

可容忍信息泄露的指定验证者签名方案

洪晓静[1*],王 斌[2]
(1.江海职业技术学院 信息工程系,江苏 扬州 225101; 2.扬州大学 信息工程学院,江苏 扬州 225127)
(*通信作者电子邮箱 hxj_net@sohu.com)

摘 要:指定验证者签名(DVS)克服了传统的数字签名中可公开验证的缺点,可防止验证者向第三方表明他获得了签名方发布的数字签名。但传统的密码方案的安全性依赖理想的假设,即攻击者不能获得保密的密钥的信息,而边信道攻击表明攻击者可以获得部分的秘密信息,因此有必要设计可以容忍信息泄露的指定验证者签名方案。基于"或"证明的技术,把 Okamoto 认证方案推广到指定验证者签名的情形,并在给定的泄露界下,证明了所提出的指定验证者签名方案在相对泄露模型下是安全的。

关键词:指定验证者签名;边信道攻击;相对泄露模型;公开验证;数字签名
中图分类号:TP309 **文献标志码**:A

Leakage-resilient designated verifier signature scheme

HONG Xiaojing[1*], WANG Bin[2]
(1. *Department of Information Engineering, Jianghai Polytechnic College, Yangzhou Jiangsu 225101, China*;
2. *College of Information Engineering, Yangzhou University, Yangzhou Jiangsu 225127, China*)

Abstract: Designated Verifier Signature (DVS) eliminates the publicly verifiable feature of traditional digital signature to prevent a verifier from proving to a third party the fact that a particular digital signature held by him is issued by a signer. Traditionally, security of digital signature schemes relies on the ideal assumption that an adversary has no access to the information of a secret key. However, side-channel attacks show that the physical implementation of a cryptographic algorithm will leak partial secret information to an adversary. Hence it is necessary to design a leakage-resilient designated verifier signature. Okamoto authentication scheme was converted into a designated verifier signature scheme by using the OR proof technique. Given the leakage bound, the security of the presented scheme was proved under the relative leakage model.

Key words: Designated Verifier Signature (DVS); side-channel attack; relative leakage model; public verifiability; digital signature

0 引言

数字签名方案允许签名人用密钥对消息签名,而任何人都可以通过签名人的公钥验证消息签名的合法性。但在某些场合下,例如关于个人健康记录等私密信息,人们不希望验证者把获得的信息和签名传递给第三方进行公开验证。为了解决这个问题,Jakobsson 等[1]提出了指定验证者签名(Designated Verifier Signature,DVS)的概念,可以防止验证者在检验了数字签名的合法性后,将消息和数字签名传递给第三方进行公开验证。这里的关键是验证者可以生成模拟签名,它与签名人发布的真实签名是不可区分的。指定验证者的数字签名在软件版权等领域有重要的应用。

但数字签名传统意义下的安全性依赖理想的假设,即攻击者不能访问签名算法内部的秘密状态(例如签名密钥或签名时使用的随机性)。然而边信道攻击(side-channel attacks)[2]表明攻击者通过算法运行时间、功耗和温度等因素可能获得内部秘密状态的部分信息。冷启动攻击(cold-boot attack)[3]表明即使在机器关机之后,攻击者仍可以从内存恢复保存在其中的密钥的部分信息。通过边信道攻击,人们发现许多在理想条件下被证明为安全的方案存在薄弱环节。而通过物理的手段来阻断所有可能的信息泄露渠道是不太现实的,所以人们开始研究在信息泄露的条件下来证明密码方案的安全性。

Micali 等[4]基于"仅有计算会泄露信息"的公设给出了模拟边信道攻击的理论框架。该公设假设攻击者只能从算法执行过程中所访问的内存获取信息。Faust 等[5]基于该框架设计了可容忍信息泄露的数字签名方案。然而该公设并不适用于基于内存和电磁辐射的边信道攻击,例如冷启动攻击等。Akavia 等[6]提出了在公钥环境下抵御边信道攻击的安全模型,要求攻击者获得的信息比特量有一个上界,也叫相对泄露模型(relative leakage model)。即攻击者可以选取一个可有效计算的任意函数 f,并能获得 $f(sk)$,sk 为密钥,但函数 f 的输出必须满足 $|f(sk)| \leqslant l$,这里 l 是一个预先给定的泄露参数。泄露参数 l 必须严格小于密钥的长度 $|sk|$,否则,由于函数 f 可以任意选取,就没有安全性可言。所以 $l/|sk|$ 也叫作相对泄露参数。相对泄露模型可以很好地模拟冷启动攻击等基于内存的泄露攻击。

Naor 等[7]基于哈希证明系统(Hash proof system)[8]设计了在相对泄露模型下可被证明为安全的公钥加密系统。Alwen 等[9]设计了容忍信息泄露的身份认证方案,并指出利

收稿日期:2013-06-13;**修回日期**:2013-08-24。
基金项目:江苏省高校自然科学研究项目(10KJD520005);扬州大学科技创新培育基金资助项目(2011CXJ022)。
作者简介:洪晓静(1977-),女,江苏姜堰人,讲师,硕士,主要研究方向:网络安全、通信技术; 王斌(1976-),男,江西萍乡人,副教授,博士,主要研究方向:公钥密码学。

用 Fiat-Shamir 变换可以导出容忍信息泄露的数字签名方案。Dodis 等[10]基于相对泄露模型给出了容忍信息泄露的数字签名的形式定义,并基于双线性映射设计了满足该定义的数字签名方案。Boyle 等[11]在连续信息泄露模型下提出了完全容忍信息泄露的数字签名方案,还允许对签名阶段使用的随机性的泄露。Dziembowski 等[12]在连续信息泄露模型下提出了自适应选择密文(Chosen Ciphertext Adaptive, CCA)安全的公钥加密方案。

由于目前的指定验证者签名方案的安全定义[13]没有涵盖信息泄露的情形,因此研究在信息泄露条件下可被证明为安全的指定验证者签名方案就很有意义。本文基于相对泄露模型定义了指定验证者签名方案的安全模型,在允许密钥信息泄露的条件下对指定验证者的签名的不可伪造性的定义进行了扩展。然后基于"或"零知识证明(OR Proof)技术和 Fiat-Shamir 变换把 Okamoto 身份认证方案[14]转换为指定验证者的数字签名方案,并使用通用分叉引理[15],在给定的泄露上界条件下,证明所提出的指定验证者签名方案是不可伪造的。

1　相关知识

用 $negl(\lambda)$ 表示可忽略的函数,即对任意的常数 c 和足够大的 λ,$negl(\lambda) \leqslant 1/\lambda^c$ 成立。令 X 代表一个随机变量。X 的极小熵(min-entropy)定义为 $H_\infty(X) = -\mathrm{lb}(\max_x \Pr[X = x])$。给定随机变量 Y 的条件下,Dodis 等[16]定义了 X 的平均极小熵 $\tilde{H}_\infty(X|Y) = -\mathrm{lb}(E_{y\leftarrow Y}[2^{-H_\infty(X|Y=y)}])$。下面的引理用于估计平均极小熵。

引理[16]　令 X,Y,Z 为随机变量,Z 分布在一个大小不超过 2^l 的集合,那么有 $\tilde{H}_\infty(X|(Y,Z)) \geqslant \tilde{H}_\infty((X,Z)|Y) \geqslant \tilde{H}_\infty(X|Y) - l$。

2　指定验证者签名方案的定义

一个指定验证者的签名方案包含以下多项式时间的算法:

1)生成参数(ParaGen):给定安全参数 λ,生成公开参数 *params*。

2)生成密钥(KeyGen):给定公开参数 *params*,生成公钥和私钥对($\boldsymbol{pk},\boldsymbol{sk}$)。

3)生成指定验证者签名(DSign):给定签名人 S 的密钥 $\boldsymbol{sk}_S$,验证者 D 的公钥 $\boldsymbol{pk}_D$ 和一条消息 M,生成一条签名 $\boldsymbol{\sigma}$。

4)验证指定验证者签名(DVer):给定验证者 D 的私钥 $\boldsymbol{sk}_D$,签名人 S 的公钥 $\boldsymbol{sk}_S$,消息 M 和对应的签名 $\boldsymbol{\sigma}$,若 $\boldsymbol{\sigma}$ 被接受则返回 1,反之则返回 0。

5)模拟指定验证者签名(DV_Sim):给定验证者 D 的私钥 $\boldsymbol{sk}_D$,签名人 S 的公钥 $\boldsymbol{sk}_S$ 和消息 M,生成一条模拟的签名 $\boldsymbol{\sigma}_{\mathrm{sim}}$。

6)签名的正确性要求:$\forall M \in \{0,1\}^*$,和 KeyGen 输出的 $(\boldsymbol{sk}_S,\boldsymbol{sk}_S)$,$(\boldsymbol{pk}_D,\boldsymbol{sk}_D)$,下面式子成立:$\boldsymbol{\sigma} \leftarrow \mathrm{DSign}(\boldsymbol{sk}_S, \boldsymbol{pk}_D, M)$,$\Pr[\mathrm{DVer}(\boldsymbol{sk}_D,\boldsymbol{sk}_S,M,\boldsymbol{\sigma}) = 1] = 1$。

3　可容忍信息泄露的 DVS 安全模型

传统上,一个指定验证者签名方案应满足:不可伪造性(unforgeability)和不可传递性(non-transferability)。而为了模拟信息泄露,攻击者可以访问一个泄露 oracle,获得关于内部秘密的部分信息。

3.1　不可伪造性

针对可以发动自适应选择消息攻击的攻击者,下面定义一个安全实验 $Exp^{\mathrm{CMA}}_{\mathrm{UF},\lambda,l}(A)$,其中攻击者 A 与一个挑战者 C 进行交互,这里 λ 代表安全参数,l 代表信息泄露的上界。

第一阶段:首先,C 运行 ParaGen(1^λ)获得 *params*;然后 C 运行 KeyGen 生成密钥对 $(\boldsymbol{sk}_S,\boldsymbol{sk}_S)$,$(\boldsymbol{pk}_D,\boldsymbol{sk}_D)$,分别为签名人和验证者的密钥对;然后令 $i \leftarrow 1$,集合 Q 初始化为空集,用于追踪攻击者 A 发出的签名查询;C 把公钥对 $(\boldsymbol{sk}_S,\boldsymbol{pk}_D)$ 和 *params* 提供给攻击者 A。

第二阶段:攻击者 A 可以向 C 发出以下查询:

1)签名查询:给定由攻击者 A 选择的一条消息 M_i,C 运行算法 DSign $(\boldsymbol{sk}_S,\boldsymbol{pk}_D,M_i)$ 获得签名 $\boldsymbol{\sigma}$,并返回 $\boldsymbol{\sigma}$ 给 A;然后 $Q \leftarrow Q \cup \{M_i\}$。

2)泄露查询:攻击者 A 选择一个比特 b 和一个可有效计算的任意函数 f_i,若 $b = 0$,C 计算 $y_i = f_i(\boldsymbol{sk}_S)$;否则计算 $y_i = f_i(\boldsymbol{sk}_D)$。令 $|y_i| = \alpha_i$,$i \leftarrow i + 1$,若 $\sum_i \alpha_i \leqslant l$,返回 y_i;否则返回 $\perp$。

第三阶段:若攻击者 A 输出消息/签名对 $(M^*,\boldsymbol{\sigma}^*)$ 满足:

1)$\mathrm{DVer}(\boldsymbol{sk}_D,\boldsymbol{sk}_S,M^*,\boldsymbol{\sigma}^*) = 1$;

2)$M^* \notin Q$;

则称 A 获胜。一个指定验证者签名方案 DVS 在 l 泄露的条件下是不可伪造的,若对任意的多项式时间的攻击者 A,在上面的实验中 A 获胜的概率 $\leqslant negl(\lambda)$。

3.2　不可传递性

不可传递性意味着模拟签名 $\boldsymbol{\sigma}_{\mathrm{sim}}$ 与真实签名 $\boldsymbol{\sigma}$ 之间是不可区分的。令 DVS 代表一个指定验证者的签名方案,针对一个区分器 D,以 λ 为安全参数,定义一个安全实验 $Exp^{\lambda}_{\mathrm{Non\text{-}Tran}}(D)$,其中 D 与一个挑战者 C 进行交互。

第一阶段:首先,C 运行 ParaGen(1^λ)获得 *params*;然后 C 运行 KeyGen 生成密钥对 $(\boldsymbol{sk}_S,\boldsymbol{sk}_S)$,$(\boldsymbol{pk}_D,\boldsymbol{sk}_D)$,分别为签名人和验证者的密钥对;$C$ 把公钥对 $(\boldsymbol{sk}_S,\boldsymbol{pk}_D)$ 和 *params* 提供给区分器 D。

区分器 D 可以向 C 发出签名查询。给定由区分器 D 选择的一条消息 M_i,C 运行算法 DSign $(\boldsymbol{sk}_S,\boldsymbol{pk}_D,M_i)$ 获得签名 $\boldsymbol{\sigma}$,并返回 $\boldsymbol{\sigma}$ 给 A;然后 $Q \leftarrow Q \cup \{M_i\}$。

挑战阶段:D 发送一条消息 $M^* \notin Q$ 给 C,然后 C 选择一个随机的 $b \in \{0,1\}$。若 $b = 0$,C 返回一条真实的签名 $\boldsymbol{\sigma}^* \leftarrow \mathrm{DSign}(\boldsymbol{sk}_S,\boldsymbol{pk}_D,M^*)$ 给 D;否则,C 返回一条模拟的签名 $\boldsymbol{\sigma}^*_{\mathrm{sim}} \leftarrow \mathrm{DV\_Sim}(\boldsymbol{sk}_D,\boldsymbol{sk}_S,M^*)$ 给 D。

猜测阶段:最终 D 输出猜测结果 b'。

如果 $b' = b$,D 获胜。D 在该实验中的优势被定义为 $Adv(D) = \left|\Pr[b' = b] - \frac{1}{2}\right|$;一个指定验证者的签名方案 DVS 是计算意义下不可传递的,若对任意的多项式时间的算法 D,有 $Adv(D) \leqslant negl(\lambda)$;一个指定验证者的签名方案 DVS 是统计意义下不可传递的,若对任意的计算能力不受限制的算法 D,有 $Adv(D) \leqslant negl(\lambda)$。

4　本文方案

给定向量 $\boldsymbol{g}$,定义一个关系 $R_{\boldsymbol{g}}$:$(y \in \mathbf{Z}_p, (x_1,x_2,\cdots,$

$x_m) \in \mathbf{Z}_p^m) \in R_g$ 当且仅当 $y = \prod_{j=1}^{m} g_j^{x_j}$。本文方案中包括以下算法：

ParaGen(1^λ)：给定安全参数 λ，令 G 为一个群，阶为素数 p，$|p| = \lambda$；然后选择 m 维的向量 $\boldsymbol{g} = (g_1, g_2, \cdots, g_m) \leftarrow_R G^m$，对参数 m 的要求将在不可伪造性的证明中给出。设 H 为一个安全的 Hash 函数，值域为 $\mathbf{Z}_p$；输出公开参数 $params = (p, G, m, \boldsymbol{g}, H)$。

KeyGen($params$)：选择密钥 $\boldsymbol{sk} = (x_1, x_2, \cdots, x_m) \leftarrow_R \mathbf{Z}_p^m$；然后设置公钥为 $\boldsymbol{pk} = \boldsymbol{g}^{sk}$，这里的向量运算的结果定义为 $\boldsymbol{pk} = \prod_{j=1}^{m} g_j^{x_j}$。

TrSim($\boldsymbol{pk}$)：接受一个公钥 $\boldsymbol{pk}$ 为输入；然后选择 $chl \leftarrow_R \mathbf{Z}_p, \boldsymbol{\sigma} = (z_1, z_2, \cdots, z_m) \leftarrow_R \mathbf{Z}_p^m$ 并计算 $cmt = \frac{(\boldsymbol{g}^{\boldsymbol{\sigma}})}{pk^{chl}}$，输出 $(cmt, chl, \boldsymbol{\sigma})$。这一步共需要 $m+1$ 步指数运算和一次求逆元运算。

DSign($\boldsymbol{sk}_S, \boldsymbol{pk}_D, M$)：令签名人 S 的密钥为 $\boldsymbol{sk}_S = (x_{S1}, x_{S2}, \cdots, x_{Sm})$，验证人 D 的公钥为 $\boldsymbol{pk}_D$，消息为 M，签名算法的步骤如下：

1) 选择 $\boldsymbol{r} = (r_1, r_2, \cdots, r_m) \leftarrow_R \mathbf{Z}_p^m$ 并计算 $cmt_S = \boldsymbol{g}^{\boldsymbol{r}}$；

2) 计算 $(cmt_D, chl_D, \boldsymbol{\sigma}_D) \leftarrow TrSim(\boldsymbol{pk}_D)$；

3) 设置 $cmt = (cmt_S, cmt_D)$ 并计算 $chl_S = H(\boldsymbol{sk}_S \| \boldsymbol{pk}_D \| cmt \| M) - chl_D$；

4) 计算 $z_{Sj} = r_j + chl_S \cdot x_{Sj} \bmod p (j = 1, 2, \cdots, m)$，令 $\boldsymbol{\sigma}_S = (z_{S1}, z_{S2}, \cdots, z_{Sm})$，输出签名 $\boldsymbol{\sigma} = (cmt_S, cmt_D, chl_S, chl_D, \boldsymbol{\sigma}_S, \boldsymbol{\sigma}_D)$。

签名算法的主要开销为 $2m+1$ 步指数运算和一次求逆元运算，但步骤 1)、2) 可以预计算。签名 $\boldsymbol{\sigma}$ 的大小为 $(2m+4)|\mathbf{Z}_p|$。

DVer($\boldsymbol{sk}_D, \boldsymbol{sk}_S, M, \boldsymbol{\sigma}$)：令验证人 D 的密钥 $\boldsymbol{sk}_D = (x_{D1}, x_{D2}, \cdots, x_{Dm})$，签名人 S 的公钥为 $\boldsymbol{pk}_S$，消息为 M 以及签名 $\boldsymbol{\sigma} = (cmt_S, cmt_D, chl_S, chl_D, \boldsymbol{\sigma}_S, \boldsymbol{\sigma}_D)$。若下面的式子通过验证则返回 1：

1) $chl_S + chl_D = H(\boldsymbol{sk}_S \| \boldsymbol{pk}_D \| cmt \| M)$；

2) $\boldsymbol{pk}_S^{chl_S} cmt_S = \boldsymbol{g}^{\boldsymbol{\sigma}_S}$；

3) $\boldsymbol{pk}_D^{chl_D} cmt_D = \boldsymbol{g}^{\boldsymbol{\sigma}_D}$。

验证算法的主要开销为 $2m+2$ 步指数运算。

DV_Sim($\boldsymbol{sk}_D, \boldsymbol{sk}_S, M$)：令验证人 D 的密钥 $\boldsymbol{sk}_D = (x_{D1}, x_{D2}, \cdots, x_{Dm})$，签名人 S 的公钥为 $\boldsymbol{pk}_S$，消息为 M。模拟签名算法的步骤如下：

1) 选择 $\boldsymbol{r}^* = (r_1^*, r_2^*, \cdots, r_m^*) \leftarrow_R \mathbf{Z}_p^m$ 并计算 $cmt_D^* = \boldsymbol{g}^{\boldsymbol{r}^*}$；

2) 计算 $(cmt_S^*, chl_S^*, \boldsymbol{\sigma}_S^*) \leftarrow TrSim(\boldsymbol{sk}_S)$；

3) 令 $cmt^* = (cmt_S^*, cmt_D^*)$ 并计算 $chl_D^* = (H(\boldsymbol{sk}_S \| \boldsymbol{pk}_D \| cmt^* \| M) - chl_S^*)$。

4) 计算 $z_{Dj}^* = r_j^* + chl_D^* \cdot x_{Dj} \bmod p, j = 1, 2, \cdots, m$。$\boldsymbol{\sigma}_D^* = (z_{D1}^*, z_{D2}^*, \cdots, z_{Dm}^*)$。

模拟签名 $\boldsymbol{\sigma}_{sim} = (cmt_S^*, cmt_D^*, chl_S^*, chl_D^*, \boldsymbol{\sigma}_S^*, \boldsymbol{\sigma}_D^*)$。

模拟签名算法的主要开销也 $2m+1$ 步指数运算和一次求逆元运算。

下面证明所提出方案的正确性，即给定签名算法 DSign 对消息 M 输出的 $\boldsymbol{\sigma} = (cmt_S, cmt_D, chl_S, chl_D, \boldsymbol{\sigma}_S, \boldsymbol{\sigma}_D)$，验证算法 DVer($\boldsymbol{sk}_D, \boldsymbol{pk}_S, M, \boldsymbol{\sigma}$) 返回 1。

由 DSign 的第 3) 步可知 $chl_S + chl_D = H(\boldsymbol{sk}_S \| \boldsymbol{pk}_D \| cmt \| M)$，即 DVer 的第 1) 步通过。

由 DSign 的第 4) 步，$\boldsymbol{\sigma}_S = (z_{S1}, z_{S2}, \cdots, z_{Sm}), z_{Sj} = r_j + chl_S \cdot x_{Sj} \bmod p (j = 1, 2, \cdots, m)$。

$$\prod_{j=1}^{m} g_j^{z_{Sj}} = \boldsymbol{g}^{\boldsymbol{\sigma}_S} = \prod_{j=1}^{m} g_j^{r_j + chl_S \cdot x_{Sj}} = \prod_{j=1}^{m} g_j^{r_j} \prod_{j=1}^{m} g_j^{chl_S \cdot x_{Sj}} = \boldsymbol{g}^{\boldsymbol{r}} \cdot \boldsymbol{pk}_S^{chl_S} = \boldsymbol{pk}_S^{chl_S} cmt_S$$

即 DVer 的第二步通过。

根据 TrSim($\boldsymbol{pk}_D$) 的计算过程，DVer 的第三步通过。

不难验证 $\boldsymbol{\sigma}_{sim}$ 也可以通过上述的验证步骤。

定理 1 给定了消息 M、签名人和验证人的公钥后，真实签名 $\boldsymbol{\sigma} = (cmt_S, cmt_D, chl_S, chl_D, \boldsymbol{\sigma}_S, \boldsymbol{\sigma}_D)$ 和模拟签名 $\boldsymbol{\sigma}_{sim} = (cmt_S^*, cmt_D^*, chl_S^*, chl_D^*, \boldsymbol{\sigma}_S^*, \boldsymbol{\sigma}_D^*)$ 是同分布的。

证明 根据 DSign 算法的执行步骤，真实签名 $\boldsymbol{\sigma}$ 的分布由相互独立的密钥 $\boldsymbol{sk}_S$，随机向量 $\boldsymbol{r} \leftarrow_R \mathbf{Z}_p^m, chl_D \leftarrow_R \mathbf{Z}_p, \boldsymbol{\sigma}_D \leftarrow_R \mathbf{Z}_p^m$ 的联合分布来决定。所以 $\boldsymbol{\sigma}$ 发生的概率如下：

$$\Pr[\boldsymbol{\sigma}] = \Pr[\boldsymbol{\sigma}_S, chl_S, (cmt_D, chl_D, \boldsymbol{\sigma}_D), cmt_S] = \Pr[\boldsymbol{\sigma}_S | chl_S, cmt_S] \cdot \Pr[chl_S, (cmt_D, chl_D, \boldsymbol{\sigma}_D), cmt_S] \tag{1}$$

式(1) 成立因为 $\boldsymbol{\sigma}_S$ 的分布只依赖签名人的公钥 $\boldsymbol{pk}_S$ 以及 chl_S 和 cmt_S。

$$\Pr[chl_S, (cmt_D, chl_D, \boldsymbol{\sigma}_D), cmt_S] = \Pr[chl_S | (cmt_D, chl_D, \boldsymbol{\sigma}_D), cmt_S] \cdot \Pr[(cmt_D, chl_D, \boldsymbol{\sigma}_D), cmt_S] \tag{2}$$

式(2) 成立因为 chl_S 的分布依赖 $(cmt_D, chl_D, \boldsymbol{\sigma}_D), cmt_S$。由相互独立性有 $\Pr[(cmt_D, chl_D, \boldsymbol{\sigma}_D), cmt_S] = \Pr[(cmt_D, chl_D, \boldsymbol{\sigma}_D)] \Pr[cmt_S]$。

下面逐步分析各式的结果。

$(cmt_D, chl_D, \boldsymbol{\sigma}_D)$ 由 TrSim($\boldsymbol{pk}_D$) 独立生成，而 TrSim 中 chl_D 是在 $\mathbf{Z}_p$ 上均匀分布，$\boldsymbol{\sigma}_D$ 在 $(\mathbf{Z}_p)^m$ 上均匀分布，给定了 $(chl_D, \boldsymbol{\sigma}_D), cmt_D$ 就完全确定了，故 $\Pr[(cmt_D, chl_D, \boldsymbol{\sigma}_D)] = 1/p^{m+1}$。$cmt_S = \boldsymbol{g}^{\boldsymbol{r}}$ 独立于 $(cmt_D, chl_D, \boldsymbol{\sigma}_D)$，在 $\mathbf{Z}_p$ 上均匀分布，$\Pr[cmt_S] = 1/p$。而固定了消息 M，签名人和验证人的公钥，以及 $cmt = (cmt_S, cmt_D)$ 和 chl_D 后，$chl_S = H(\boldsymbol{sk}_S \| \boldsymbol{pk}_D \| cmt \| M) - chl_D$ 的值就被完全确定了，所以 $\Pr[chl_S | (cmt_D, chl_D, \boldsymbol{\sigma}_D), cmt_S] = 1$。

下面在固定签名人的公钥 $\boldsymbol{sk}_S$ 以及 chl_S 和 cmt_S 的条件下，分析 $\boldsymbol{\sigma}_S = (z_{S1}, z_{S2}, \cdots, z_{Sm})$ 的概率分布，其中 $z_{Sj} = r_j + chl_S \cdot x_{Sj} \bmod p (j = 1, 2, \cdots, m)$。

$\boldsymbol{\sigma}_S$ 的分布取决于向量 $\boldsymbol{r}$ 与密钥 $\boldsymbol{sk}_S$。可以列出矩阵方程如下：

$$\begin{bmatrix} 1 & chl_S & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & chl_S & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} r_1 \\ x_{S1} \\ \vdots \\ r_m \\ x_{Sm} \end{bmatrix} = \begin{bmatrix} z_{S1} \\ z_{S2} \\ \vdots \\ z_{Sm} \end{bmatrix}$$

由该 m 行 $2m$ 列的矩阵可以确定能生成 $\boldsymbol{\sigma}_S$ 的向量 $\boldsymbol{r}$ 与密钥 $\boldsymbol{sk}_S$ 的组合共有 p^m 种。

由有限域上线性代数的知识，固定签名人的公钥 $\boldsymbol{pk}_S$、cmt_S，向量 $\boldsymbol{r}$ 与密钥 $\boldsymbol{sk}_S$ 的组合共有 $p^{2(m-1)}$ 种，故此时 $\Pr[\boldsymbol{\sigma}_S | chl_S, cmt_S] = p^m/(p^{m-1} \cdot p^{m-1}) = 1/p^{m-2}$。

所以 $\boldsymbol{\sigma}$ 发生的概率为 $1/p^{2m}$。

DV_Sim 算法和 DSign 算法仅是交换了签名人和验证人

的角色,不难计算 σ_{sim} 发生的概率也为 $1/p^{2m}$。

通用的分叉引理(General Forking Lemma)[3]:固定一个整数 $q \geqslant 1$ 和一个集合 H 使得 $|H| \geqslant 2$。令 A 为一个随机算法,输入为 $(x,h_1,h_2,\cdots,h_q)$,输出为两个元素,其中第一个元素是 $\{0,1,\cdots,q\}$ 中的一个数,输出的第二项内容取决于具体的应用。令 $Coins_A$ 代表 A 进行随机性选择的集合。设算法 IG 为 A 提供输入 x,则 A 的接受概率定义如下:

$$acc = \Pr[I \geqslant 1 : x \xleftarrow{\$} IG; (h_1,h_2,\cdots,h_q) \in_R H; (I,\sigma) \leftarrow A(x,h_1,h_2,\cdots,h_q)]$$

以 x 为输入,分叉算法 $F_A(x)$ 定义如下:

算法 $F_A(x)$
$\rho \leftarrow_R Coins_A; (h_1,h_2,\cdots,h_q) \leftarrow_R H$
$(I,\sigma) \leftarrow A(x,h_1,h_2,\cdots,h_q;\rho)$
若 $I = 0$,返回 $(0,\varepsilon,\varepsilon)$
$(h_1',\cdots,h_q') \leftarrow_R H$
$(I',\sigma') \leftarrow A(x,h_1,\cdots,h_{I-1},h_I',\cdots,h_q';\rho)$
若 $(I = I' \wedge h_I \neq h_I')$,返回 $(1,\sigma,\sigma')$,否则返回 $(0,\varepsilon,\varepsilon)$

F_A 的成功概率定义为 $frk = \Pr[b = 1 : x \xleftarrow{\$} IG; (b,\sigma,\sigma') \leftarrow F_A(x)]$。

那么 $frk \geqslant acc \cdot \left(\frac{acc}{q} - \frac{1}{|H|}\right)$ 成立。

定理 2　设安全参数为 λ,m 为本文 DVS 方案中的公开参数,且泄露参数 $l \leqslant \frac{(m-1)\mathrm{lb}\,p - \omega(\mathrm{lb}\,\lambda)}{2}$。若有攻击者 A 在安全实验 $Exp^{CMA}_{UF,\lambda,l}(A)$ 中至少以概率 ε 攻破本文的 DVS 方案,且 A 向 random oracle H 发出的查询次数为 q_H,那么离散对数问题可以至少以概率 $\frac{\varepsilon}{m} \cdot \left(\frac{\varepsilon}{q_H} - \frac{1}{p}\right)$ 被求解。

证明　给定一个素数阶 p 的群 G,生成元为 g,离散对数求解器 B 获得 g^x 作为输入。B 随机选择一个下标 $\rho \in [1,m]$,m 作为本文 DVS 方案中的公开参数;然后定义一个 m 维向量 $\boldsymbol{g}$,各分量为:$g_\rho = g^x, g_i = g^{a_i}, a_i \leftarrow \mathbf{Z}_p (i \in [1,m], i \neq \rho)$。公开参数为 $params = (p,G,m,\boldsymbol{g},H)$;然后 B 选择 $\{h_1,h_2,\cdots,h_{q_H}\} \leftarrow_R \mathbf{Z}_p$,以及 $\rho_A \leftarrow_R Coins_A$ 和 ρ_K。

第一阶段:B 运行一个模拟器 $Sim(params,\{h_1,h_2,\cdots,h_{q_H}\};\rho_K \| \rho_A)$。模拟器 Sim 按如下方式运行:

Sim 首先运行 KeyGen$(params;\rho_K)$ 获得密钥对 $(\boldsymbol{sk}_S,\boldsymbol{sk}_S),(\boldsymbol{pk}_D,\boldsymbol{sk}_D)$,分别为签名人和验证者的密钥对。然后 Sim 运行算法 $A((\boldsymbol{sk}_S,\boldsymbol{pk}_D);\rho_A)$ 并与 A 按如下方式进行交互:

1) Sim 使用密钥 $\boldsymbol{sk}_S$ 来回答 A 发出的签名查询和泄露查询。

2) 当 A 发出第 i 次 random oracle 查询 $Q_i = (\boldsymbol{sk}_S \| \boldsymbol{pk}_D \| cmt_i \| M_i)$ 时,Sim 返回预设的 h_i 作为响应。

最终 A 输出消息/签名对 (M',σ'),$\sigma' = (cmt_S', cmt_D', chl_S', chl_D', \sigma_S', \sigma_D')$。若存在某个 random oracle 查询 Q_J,查询下标为 J,满足 $Q_J = (\boldsymbol{sk}_S \| \boldsymbol{pk}_D \| (cmt_S', cmt_D') \| M')(1 \leqslant J \leqslant q_H)$,并且 A 在安全实验 $Exp^{CMA}_{UF,\lambda,l}(A)$ 中获胜,则 Sim 返回 $(J,(M',\sigma'))$ 给 B;否则 Sim 返回 $(0,(M',\sigma'))$ 给 B。

定义 $acc = \Pr[1 \leqslant J \leqslant q_H \wedge A \text{ wins}]$。由于 Hash 函数被模拟为 random oracle,若 $J \notin [1,q_H]$,A 在安全实验 $Exp^{CMA}_{UF,\lambda,l}(A)$ 中获胜的概率不超过 $1/p$,因此 $acc = \Pr[A \text{ wins}] - (1/p) \approx \varepsilon$。若 $J \neq 0$,B 进入到第二阶段;否则 B 返回 $(0,\varepsilon,\varepsilon)$ 并退出。

第二阶段:B 保持公开参数 $params$ 和上一阶段的 ρ_K,ρ_A 不变,并重新选择 $\{h_J',\cdots,h_{q_H}'\} \leftarrow_R \mathbf{Z}_p$,然后 B 重新运行模拟器:$Sim(params,\{h_1,h_2,\cdots,h_{J-1},h_J',\cdots,h_{q_H}'\};\rho_K \| \rho_A)$。

模拟器 Sim 按上一阶段同样的规则运行算法 $A((\boldsymbol{sk}_S,\boldsymbol{pk}_D);\rho_A)$ 并与之进行交互。

设在第二阶段 A 最终输出消息/签名对 (M'',σ''),$\sigma'' = (cmt''_S, cmt_D'', chl''_S, chl_D'', \sigma''_S, \sigma_D'')$。若存在某个 random oracle 查询 $Q_{J'}$,查询下标为 J',满足 $Q_{J'} = (\boldsymbol{sk}_S \| \boldsymbol{pk}_D \| (cmt''_S, cmt_D'') \| M'')(1 \leqslant J' \leqslant q_H)$,并且 A 在安全实验 $Exp^{CMA}_{UF,\lambda,l}(A)$ 中获胜,Sim 返回 $(J',(M'',\sigma''))$ 给 B;否则 Sim 返回 $(0,(M'',\sigma''))$ 给 B。

此时设 B 获得了成功的分叉,即有 $J = J'$ 且 $h_J \neq h_J'$,用 E 代表获得成功分叉的事件。根据通用的分叉引理,事件 E 发生的概率为 $frk \geqslant \varepsilon \cdot \left(\frac{\varepsilon}{q_H} - \frac{1}{p}\right)$。在成功分叉条件下,在第二阶段 A 发出的第 J 次 random oracle 查询 $Q_{J'}$ 和第一阶段发出的第 J 次 random oracle 查询的内容相同,因为它们都由 $(h_1,h_2,\cdots,h_{J-1})$ 以及固定的随机性 ρ_A 决定。因此有下面的方程:

$$(cmt_S', cmt_D') = (cmt''_S, cmt_D''), M' = M''$$

而根据验证方程有 $chl_S' + chl_D' = h_J, chl''_S + chl_D'' = h_J'$,$h_J \neq h_J'$。

此时又可以分以下两种情形讨论:

1) 用 E_1 代表事件 $chl_S' = chl''_S$ 发生:此时有 $chl_D' \neq chl_D''$。根据 DVer 算法的第三个验证式子,有 $pk_D^{chl_D'} cmt_D' = \boldsymbol{g}_D^{\sigma'} pk_D^{chl_D''} cmt_D'' = \boldsymbol{g}_D^{\sigma''}$。

即有

$$\boldsymbol{pk}_D^{(chl_D' - chl_D'')} = \boldsymbol{g}_D^{(\sigma_D' - \sigma'')} = \prod_{j=1}^{m} g_j^{(\sigma_{Dj}' - \sigma_{Dj}'')}$$

$$\boldsymbol{pk}_D = \prod_{j=1}^{m} g_j^{(\sigma_{Dj}' - \sigma''_{Dj}) \cdot (chl_D' - chl_D'')^{-1}}$$

换句话说,B 可以抽取 $\boldsymbol{sk}_D'$ 使 $(\boldsymbol{pk}_D,\boldsymbol{sk}_D')$ 属于关系 R_g:

$$\boldsymbol{sk}_D' = ((\sigma_{D1}' - \sigma''_{D1}) \cdot (chl_D' - chl_D'')^{-1}, \cdots, (\sigma_{Dm}' - \sigma_{Dm}'') \cdot (chl_D' - chl_D'')^{-1})$$

令 E_{11} 代表事件 $\boldsymbol{sk}_D' \neq \boldsymbol{sk}_D$,这里 $\boldsymbol{sk}_D$ 为由模拟器 Sim 运行 KeyGen 事先生成的验证者密钥。由于签名查询是用签名人密钥 $\boldsymbol{sk}_S$ 回答的,签名查询不提供验证者密钥 $\boldsymbol{sk}_D$ 的信息。因此在上述过程中 A 只能从验证者公钥 $\boldsymbol{pk}_D$ 和泄露查询中获取关于验证者密钥 $\boldsymbol{sk}_D$ 的信息。令 SK,PK,Z 为三个随机变量分别表示密钥 $\boldsymbol{sk}_D$,公钥 $\boldsymbol{pk}_D$ 以及全部关于密钥 $\boldsymbol{sk}_D$ 泄露信息的分布。

根据引理 1,有 $H_\infty(SK|(PK,Z)) \geqslant H_\infty((SK,Z)|PK) \geqslant H_\infty(SK|PK) - 2l$。

需要注意的是 B 运行 A 共两次,每次 A 都可以发出泄露查询,每次 A 全部泄露查询的界为 l,所以最终全部泄露信息的界最多为 $2l$。

由定理 1 的分析有 $H_\infty(SK|PK) = (m-1)\mathrm{lb}\,p$。

那么推出 $\Pr[\boldsymbol{sk}_D' = \boldsymbol{sk}_D] \leqslant 2^{-H_\infty(SK|(PK,Z))} \leqslant 2^{2l - H_\infty(SK|PK)}$。

此时要求 $H_\infty(SK|PK) - 2l$ 满足关于 λ 是超对数阶的增长,即 $H_\infty(SK|PK) - 2l \geqslant \omega(\mathrm{lb}\,\lambda)$。为满足上式,要求泄露参数 $l \leqslant \frac{(m-1)\mathrm{lb}\,p - \omega(\mathrm{lb}\,\lambda)}{2}$。那么有

$$\Pr[E_1 \wedge \boldsymbol{sk}_D' \neq \boldsymbol{sk}_D] \geqslant \Pr[E_1] - \Pr[\boldsymbol{sk}_D' = \boldsymbol{sk}_D] = \Pr[E_1] - 2^{-\omega(\mathrm{lb}\,\lambda)} = \Pr[E_1] - negl(\lambda)$$

2) 令 E_2 代表事件 $chl_D' = chl_D''$ 发生,此时有 $chl_S' \neq$

chl''_S。类似地，推出 B 可以抽取 $\boldsymbol{sk}_S'$ 使 $(\boldsymbol{sk}_S, \boldsymbol{sk}_S')$ 属于关系 R_g：

$$\boldsymbol{sk}_S' = ((\boldsymbol{\sigma}_{S1}' - \boldsymbol{\sigma}_{S1}'') \cdot (chl_S' - chl_S'')^{-1}, \cdots, (\boldsymbol{\sigma}_{Sm}' - \boldsymbol{\sigma}_{Sm}'') \cdot (chl_S' - chl_S'')^{-1})$$

令 E_{21} 代表事件 $\boldsymbol{sk}_S' \neq \boldsymbol{sk}_S$，这里 $\boldsymbol{sk}_S$ 为由模拟器 Sim 运行 KeyGen 事先生成的签名人密钥。由于根据定理 1 签名查询可以使用验证者密钥 $\boldsymbol{sk}_D$ 通过 DV_Sim(·) 完美地进行模拟，签名查询不提供密钥 $\boldsymbol{sk}_S$ 的信息。因此在上述过程中 A 只能从签名人公钥 $\boldsymbol{sk}_S$ 和泄露查询中获取关于签名人密钥 $\boldsymbol{sk}_S$ 的信息。令 SK、PK、Z 为三个随机变量分别表示密钥 $\boldsymbol{sk}_S$，公钥 $\boldsymbol{sk}_S$ 以及全部关于密钥 $\boldsymbol{sk}_S$ 泄露信息的分布。

类似地，有 $H_\infty(SK|(PK,Z)) \geqslant H_\infty((SK,Z)|PK) \geqslant H_\infty(SK|PK) - 2l$，以及 $H_\infty(SK|PK) = (m-1)\text{lb}\, p$ 成立。

那么推出 $\Pr[\boldsymbol{sk}_S' = \boldsymbol{sk}_S] \leqslant 2^{-H_\infty(SK|(PK,Z))} \leqslant 2^{2l-H_\infty(SK|PK)}$。

此时要求 $H_\infty(SK|PK) - 2l$ 满足关于 λ 是超对数阶的增长，即 $H_\infty(SK|PK) - 2l \geqslant \omega(\text{lb}\,\lambda)$。

为满足上式，要求泄露参数 $l \leqslant \dfrac{(m-1)\text{lb}\, p - \omega(\text{lb}\,\lambda)}{2}$。

那么有

$$\Pr[E_2 \wedge \boldsymbol{sk}_S' \neq \boldsymbol{sk}_S] \geqslant \Pr[E_2] - \Pr[\boldsymbol{sk}_S' = \boldsymbol{sk}_S] = \Pr[E_2] - 2^{-\omega(\text{lb}\,\lambda)} = \Pr[E_2] - negl(\lambda)$$

综上所述，只要发生上述两种情况之一，B 都可以获得 $\boldsymbol{pk}, \boldsymbol{sk}, \boldsymbol{sk}'$ 满足 $\boldsymbol{sk} \neq \boldsymbol{sk}'$ 且 $(\boldsymbol{pk}, \boldsymbol{sk}), (\boldsymbol{pk}, \boldsymbol{sk}') \in$ 关系 R_g，概率为 $\Pr[E_1] + \Pr[E_2] - negl(\lambda) \approx \Pr[E_1] + \Pr[E_2] = frk$。

现在令 $\boldsymbol{sk} = (x_1, x_2, \cdots, x_m)$，$\boldsymbol{sk}' = (x_1', x_2', \cdots, x_m')$。若 $\boldsymbol{sk} \neq \boldsymbol{sk}'$，则在第 ρ 个分量出 $x_\rho \neq x_\rho'$ 以概率 $\dfrac{1}{m}$ 成立。在 $x_\rho \neq x_\rho'$ 的条件下，有下面的式子：

$$g_{i=1,i\neq\rho}^{\sum^m a_i x_i} (g^x)^{x_\rho} = g_{i=1,i\neq\rho}^{\sum^m a_i x_i'} (g^x)^{x_\rho'} \Rightarrow$$

$$\sum_{i=1,i\neq\rho}^{m} a_i x_i + x \cdot x_\rho \equiv \sum_{i=1,i\neq\rho}^{m} a_i x_i' + x \cdot x_\rho' \bmod p \Rightarrow$$

$$x = \sum_{i=1,i\neq\rho}^{m} a_i (x_i' - x_i) \cdot (x_\rho - x_\rho')^{-1} \bmod p$$

最终推出 B 求解离散对数成功的概率为 $\dfrac{\varepsilon}{m} \cdot \left(\dfrac{\varepsilon}{q_H} - \dfrac{1}{p}\right)$。

5 结语

本文首先基于相对泄露模型，设计了可容忍信息泄露的、指定验证者签名方案的安全模型。在允许密钥信息泄露的条件下，对指定验证者签名的不可伪造性的定义进行了扩展。然后基于“或”的零知识证明技术和 Fiat-Shamir 变换把 Okamoto 身份认证方案转换为指定验证者的数字签名方案。在给定的泄露上界下，最后证明了所提出的方案可以满足安全模型中不可伪造性的定义，且具有完美的不可传递性。

参考文献：

[1] JAKOBSSON M, SAKO K, IMPAGLIAZZO R. Designated verifier proofs and their applications [C]// EUROCRYPT '1996: Proceedings of the 15th Annual International Conference on Theory and Application of Cryptographic Techniques, LNCS 1070. Berlin: Springer-Verlag, 1996: 143 – 154.

[2] KOCHER P C. Timing attacks on the implementations of Diffie-Hellman, RSA, DSS, and other systems [C] // CRYPTO '1996: Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology. Berlin: Springer-Verlag, 1996: 104 – 113.

[3] HALDERMAN J A, SCHOEN S D, HENINGER N, *et al.* Lest we remember: cold boot attacks on encryption keys [C]// Proceedings of 17th USENIX Security symposium. Berkeley: USENIX Association, 2008: 45 – 60.

[4] MICALI S, REYZIN L. Physically observable cryptography [C] // Proceedings of Theory of Cryptography'2004, LNCS 2951. Berlin: Springer-Verlag, 2004: 278 – 296.

[5] FAUST S, KILTZ E, PIETRZAK K, *et al.* Leakage-resilient signatures [C] // Proceedings of Theory of Cryptography'2010, LNCS 5978. Berlin: Springer-Verlag, 2010: 343 – 360.

[6] AKAVIA A, GOLDWASSER S, VAIKUNTANATHAN V. Simultaneous hardcore bits and cryptography against memory attacks [C] // Proceedings of Theory of Cryptography'2009, LNCS 5444. Berlin: Springer-Verlag, 2009: 474 – 495.

[7] NAOR M, SEGEV G. Public-key cryptosystems resilient to key leakage [C]// CRYPTO'2009: Proceedings of Advance in Cryptology, LNCS 5677. Berlin: Springer-Verlag, 2009: 18 – 35.

[8] CRAMER R, SHOUP V. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption [C] // Proceedings of Advance in Cryptology-EUROCRYPT'2002, LNCS 2332. Berlin: Springer-Verlag, 2002: 45 – 64.

[9] ALWEN J, DODIS Y, NAOR M, *et al.* Public-key encryption in the Bounded Retrieval Model [C] // Proceedings of Advance in Cryptology- EUROCRYPT'2010, LNCS 6110. Berlin: Springer-Verlag, 2010: 113 – 134.

[10] DODIS Y, HARALAMBIEV K, LOPEZ-ALT A, *et al.* Efficient public-key cryptography in the presence of key leakage [C] // Proceedings of Advance in Cryptology-ASIACRYPT'2010, LNCS 6477. Berlin: Springer-Verlag, 2010: 613 – 631.

[11] BOYLE E, SEGEV G, WICHS D. Fully leakage-resilient signatures [C] // Proceedings of Advance in Cryptology-EUROCRYPT'2011, LNCS 6632. Berlin: Springer-Verlag, 2011: 89 – 108.

[12] DZIEMBOWSKI S, FAUST S. Leakage-resilient cryptography from the inner-product extractor [C] // Proceedings of Advance in Cryptology-ASIACRYPT'2011, LNCS 7073. Berlin: Springer-Verlag, 2011: 702 – 721.

[13] LI Y, SUSILO W, MU Y, *et al.* Designated verifier signature: definition, framework and new constructions [C] // Proceedings of Ubiquitous Intelligence and Computing'2007, LNCS 4611. Berlin: Springer-Verlag, 2007: 1191 – 1200.

[14] OKAMOTO T. Provably secure and practical identification schemes and corresponding signature schemes [C] // CRYPTO'1992: Proceedings of Advance in Cryptology, LNCS 740. Berlin: Springer, 1992: 31 – 53.

[15] BELLARE M, NEVEN G. Multi-signatures in the plain public-key model and a general forking lemma [C] // Proceedings of 13th ACM Conference on Computer and Communications Security. New York: ACM, 2006: 390 – 399.

[16] DODIS Y, OSTROVSKY R, REYZIN L, *et al.* Fuzzy extractors: How to generate strong keys from biometrics and other noisy data [J]. SIAM Journal on Computing, 2008, 38(1): 97 – 139.