论 文

基于可编程 hash 函数的短签名

王志伟^{①②③④}^① 南京邮电大学计算机学院, 南京 210003^② 信息安全国家重点实验室中国科学院信息工程研究所, 北京 100093^③ 江苏省无线传感网高技术研究重点实验室, 南京 210003^④ 宽带无线通信与传感网技术教育部重点实验室, 南京 210003

E-mail: zhwwang@njupt.edu.cn

收稿日期: 2011-08-12; 接受日期: 2011-10-31

国家自然科学基金 (批准号: 61272422, 61202353)、江苏高校优势学科建设工程 (批准号: yx002001) 和江苏省高校自然科学基金 (批准号: 11KJB520015) 资助项目

摘要 数字签名中的短签名由于其签名长度的优势, 特别适用于通信带宽受限的场合. 现有的短签名方案大多是随机预言模型下可证明安全的, 但是随机预言模型通常被认为过于理想化, 现实中没有一种 hash 函数能够模拟随机预言模型, 而少数标准模型下可证安全的短签名方案, 一般被认为是低效的或者基于强困难假设, 即攻击者被给予一定数量的随机的已解决问题实例, 要求去解决一个它自己选择的实例. 可编程 hash 函数 (programmable hash functions, PHF) 是一种能模拟随机预言的某些可编程特性的特殊 hash 函数. 可编程 hash 函数可嵌入到签名的基本构造中, 产生标准模型下的短签名. 本文利用可编程 hash 函数设计了一个基于因子分解假设的短签名方案. 它具有的优点是: 1) 签名长度短, 只需要一个群上的元素和一个小整数; 2) 签名和验证计算量小, 不需要在签名过程中进行生成素数的运算; 3) 不需要嵌入变色龙 hash 函数便可实现标准模型下可证明安全.

关键词 短签名 密码学 因子分解假设 可编程 hash 函数 可证明安全

1 引言

数字签名是一种最重要的密码学原语之一, 被应用于构造各种高层次的密码协议. 对数字签名的研究和对公钥密码的研究是同时开始的. 1976 年, 文献 [1] 中 Diffie 和 Hellman 首次提出了公钥密码系统的概念, 并且提出了在公钥密码系统中实现数字签名的思想, 但是并没有给出具体的方案. 第一个数字签名方案是由 Rivest, Shamir 和 Adleman 三位密码学者首先提出的, 即著名的 RSA 签名 [2]. 此后, 数字签名理论与技术在密码学界受到了广泛的重视. 数字签名中的短签名由于其签名长度的优势, 特别适用于通信带宽受限的场合, 如无线传感网、物联网等. 但是, 目前实用的短签名方案 [3,4] 安全性大多基于随机预言模型 [5], 而随机预言模型通常被认为过于理想化 [6].

现有的标准模型下可证安全的短签名方案, 通常被认为是低效的或者基于强困难假设 (强 $-q$ 假设). 在强 $-q$ 假设中, 攻击者被给予一定数量的随机的已解决问题实例, 要求去解决一个它自己选择的实例. 例如, 文献 [7~10] 是基于强 RSA 假设, 文献 [10,11] 是基于强 $-q$ -Diffie-Hellman 假设. 在标准

引用格式: 王志伟. 基于可编程 hash 函数的短签名. 中国科学: 信息科学, 2013, 43: 335-342, doi: 10.1360/112011-874

困难假设中, 攻击者必须解决一个给定的固定实例. 用标准困难假设去构造短签名, 难度比较大. 例如标准 RSA 假设, 目前仅有 Hohenberger 等^[12]提出的一个短签名方案, 以及 Hofheinz 等^[13]提出的一个基于可编程 hash 函数的短签名方案. 但是, 他们的签名都需要大量的素数生成运算, 计算效率比较低. Hofheinz 等^[13]提出了一个公开问题, 即如何构造一个实用的基于标准 RSA 或者因子分解假设的短签名, 签名过程中不需要素数生成运算.

可编程 hash 函数由 Hofheinz 等^[9,13]提出, 可模拟随机预言的某些性质, 对签名的构造很有帮助. 本文利用 $(m, 1)$ - 可编程 hash 函数^[13]设计一个基于因子分解假设的短签名方案, 其标准模型下的可证明安全性不再需要嵌入变色龙 hash 函数^[12]. 签名长度仅为一个 $\mathbb{Z}_N$ 上的元素和一个小整数, 和 Hofheinz RSA 签名^[13]长度相当. 而且, 签名过程不再需要素数生成运算, 计算过程变得高效实用. 本方案也回答了 Hofheinz 等提出的公开问题.

2 相关概念

2.1 数学基础

本节对一些数学中的基本概念和相关定理作一个介绍.

二次剩余: 给定正整数 N , 如果整数 a 满足 $\text{GCD}(a, N) = 1$, 并且存在整数 x 使得 $x^2 \equiv a \pmod{N}$, 那么 a 称为整数模 N 下的平方根. $N = p \times q$ 是一个合数, 其中 p 和 q 是两个素数. Q_N 是乘法群 $\mathbb{Z}_N^*$ 中所有二次剩余组成的子群.

因子分解问题: 给定一个合数 N 是两个安全大素数 $p = 2p' + 1$ 和 $q = 2q' + 1$ 的乘积, 输出 p 或 q .

定理 2.1^[14] 令 $N = p \times q$, 其中 $p = 2p' + 1$ 和 $q = 2q' + 1$ 是安全大素数, p' 和 q' 也是大素数, 若给出 $a \in Q_N$, 则满足 $a^{2^d} \equiv a \pmod{N}$, 其中 $d = (N - p - q + 5)/8$.

定理 2.1 提供了一个计算二次剩余 $a \in Q_N$ 的平方根的方法, 即计算 a^d .

定理 2.2^[14] 令 $N = p \times q$, 其中 $p = 2p' + 1$ 和 $q = 2q' + 1$ 是安全大素数, p' 和 q' 也是大素数, 令 $d = (N - p - q + 5)/8$, 若 $a = x^2 \in Q_N$, 那么 $a^d \in Q_N$.

由定理 2.2 可知 a^d 仍然是二次剩余, 可以对 $a \in Q_N$ 进行 λ 次 d 幂运算, 求得 a 的 2^λ 次根 ($\varphi(N) = (p-1)(q-1) \geq 2^l$), 而不用担心中间出现二次非剩余.

下面介绍一个安全证明中要用的引理.

引理 2.1^[13] 令 A 是个集合, 大小为 $|A|$. 令 $X_1, \dots, X_q$ 是 A 上随机选取的 q 个独立的随机变量, 则存在 $m+1$ 个下标 $i_1, \dots, i_{m+1}$ 满足 $X_{i_1} = \dots = X_{i_{m+1}}$ 的概率上限为 $\frac{q^{m+1}}{a^m}$.

2.2 可编程 hash 函数的定义

可编程 hash 函数是一种可以模拟随机预言某些性质的特殊 hash 函数^[9,13], 可直接嵌入签名的构造之中. 下面我们介绍它的定义.

一个可编程 hash 函数 PHF 作用在群 G 上, 输入长度为 $l = l(k)$ (k 是安全参数), 由两个高效的算法 PHF.Gen 和 PHF.Eval 组成. 概率算法 PHF.Gen(1^k) 产生 hash 密钥 κ , 确定算法 PHF.Eval 输入 hash 密钥 κ 和 $X \in \{0, 1\}^l$, 输出 $\text{PHF.Eval}(\kappa, X) \in G$.

可编程 hash 函数: 一个可编程 hash 函数 $\text{PHF} = (\text{PHF.Gen}, \text{PHF.Eval})$ 称为是 (m, n, γ, δ) 可编程的, 当且仅当存在一个陷门密钥生成算法 PHF.TrapGen 和一个陷门计算算法 PHF.TrapEval 满足

下列性质:

1) 陷门密钥生成算法 $\text{PHF.TrapGen}(1^k, g, h)$ 输入两个 G 上的生成元 g, h , 输出 hash 密钥 κ 和陷门信息 τ .

2) 对于 G 上所有的生成元 g, h , 由 $\text{PHF.Gen}(1^k)$ 生成的 hash 密钥 κ 和由 $\text{PHF.TrapGen}(1^k, g, h)$ 生成的密钥 κ' 是统计学上 γ 接近的.

3) 输入 $X \in \{0, 1\}^l$ 和陷门信息 τ , 陷门计算算法 $\text{PHF.TrapEval}(\tau, X)$ 输出一对整数 $a_X, b_X \in \mathbb{Z}$. 对于所有 $X \in \{0, 1\}^l$, 确定算法 PHF.Eval 有如下输出结果:

$$\text{PHF.Eval}(\tau, X) = g^{a_X} h^{b_X}.$$

4) 对于 G 上所有的生成元 g, h , 所有由 $\text{PHF.TrapGen}(1^k, g, h)$ 产生的 hash 密钥 κ , 以及所有 $X_1, \dots, X_m \in \{0, 1\}^l$ 和 $Z_1, \dots, Z_n \in \{0, 1\}^l$ 满足 $X_i \neq Z_j$, 我们有

$$\Pr[a_{X_1} = \dots = a_{X_m} = 0 \quad \text{and} \quad a_{Z_1} = \dots = a_{Z_n} \neq 0] \geq \delta,$$

其中 $(a_{X_i}, b_{X_i}) = \text{PHF.TrapEval}(\tau, X_i)$ 和 $(a_{Z_j}, b_{Z_j}) = \text{PHF.TrapEval}(\tau, Z_j)$, 概率 δ 的计算依赖于陷门信息 τ 和 hash 密钥 κ .

如果 γ 是可忽略的, 而 δ 值比较大, 则 PHF 被称为是 (m, n, γ, δ) 可编程 hash 函数.

3 方案的设计

我们的构造基于 Hofheinz 等^[13]的 $(m, 1)$ -可编程 hash 函数. 令 $l = l(k)$ 和 $\lambda = \lambda(k)$ 是多项式, k 是安全参数. 令 $\text{PHF} = (\text{PHF.Gen}, \text{PHF.Eval})$ 是一个作用在二次剩余群 Q_N 上的可编程 hash 函数, 输入长度是 l . 短签名方案由 3 个算法组成: KeyGen, Sign, Verify.

KeyGen (1^k) : 算法选取一个合数 N , 是两个安全大素数 $p = 2p' + 1$ 和 $q = 2q' + 1$ 的乘积, 满足 $(p-1)(q-1) \geq 2^\lambda$ 和 $pq < 2^{\lambda+1}$. 计算 $d = (N - p - q + 5)/8$. 接下来用概率算法 $\text{PHF.Gen}(1^k)$ 产生一个 hash 密钥 κ . 为伪随机函数族函数 $F: \{0, 1\}^* \rightarrow \{0, 1\}^\lambda$ 选取一个密钥 K 和随机数 $c \in \{0, 1\}^\lambda$. 构造 hash 函数 $H_K(x) := c \oplus F_K(x)$. $H_K(x)$ 不同于一般意义上的 hash 函数, 因为密钥 K 和随机数 c 是公开的, 因此 $H_K(x)$ 的计算也是公开的^[12]. 公钥为 $\text{PK} = (N, \kappa, K, c)$, 私钥 $\text{SK} = d$. (为简单起见, 下面我们将 $\text{PHF.Eval}(\tau, M)$ 写成 $H(M)$).

Sign (SK, M) : 输入私钥 SK 和消息 $M \in \{0, 1\}^l$, 算法随机选择 $s \in \{0, 1\}^\lambda$. 用 $s_{|i}$ 表示 s 前 i 个比特组成的前缀, 计算 $P(s) = \sum_{i=1}^\lambda H_K(s_{|i})$, 输出签名为

$$\sigma = H(M)^{d^{P(s)}} \bmod N.$$

签名是 $(\sigma, s) \in \mathbb{Z}_N \times \{0, 1\}^\lambda$.

Verify $(\text{PK}, M, (\sigma, s))$: 输入公钥 PK , 消息 M 和签名 (σ, s) , 验证等式

$$\sigma^{2^{P(s)}} = H(M) \bmod N,$$

是否成立. 如果等式成立, 则输出“签名有效”, 否则输出“签名无效”.

Verify 算法的正确性: 由定理 2.1 可知, 若 $a \in Q_N$ 且 $d = (N - p - q + 5)/8$, 则 $a^d = a^{1/2} \bmod N$. 因此 $\sigma = H(M)^{d^{P(s)}} = H(M)^{(1/2)^{P(s)}} \bmod N$, 则验证方程 $\sigma^{2^{P(s)}} = H(M) \bmod N$ 成立.

效率分析: 上面的签名只需要一个 $\mathbb{Z}_N$ 上的元素和一个小整数, 签名长度很短, 节约了通信带宽. 在计算效率方面, HW 签名方案和 Hofheinz 的 RSA 签名都要求伪随机函数构造的函数输出是奇素数时, 才能被采用, 因此在签名算法和验证算法中存在大量的素数生成和素性检测计算. 相比之下, 我们的方案中, $H_K(X)$ 的输出不再需要必须是奇素数, 因而省却了大量的计算. 我们签名过程只需要 2 次模指数运算, 验证过程也很高效, 因为计算 $\sigma^{2^{P(s)}}$ 只需要加法和移位运算. 不仅如此, 和 HW 签名方案签名^[12] 相比, 我们签名不需要嵌入变色龙 hash 函数便能实现标准模型下的可证安全, 这也大大提高了效率.

4 安全性证明

定理 4.1 设 PHF 是一个 $(m, 1, \gamma, \delta)$ - 可编程 hash 函数. 如果因子分解问题是困难的, 那么 3 节的签名方案是安全的.

证明 为了证明上述定理, 我们需要区分两种类型的攻击者 I 和 II. I 型攻击者伪造的签名 (M^*, σ^*, s^*) 满足 $s^* = s_i$ ($i \in \{1, \dots, q\}$), 也就是说 I 型攻击者重用了签名查询过程中的某个随机值 s_i . II 型攻击者伪造的签名则是选用了一个新的 s^* .

关于 I 型攻击者的证明过程需要说明的是, 由于存在一个或多个 $i \in [q]$ 使得 $s^* = s_i$, 挑战者 C 不得不准备多个形如 $H(M_i)^{(1/2)^{P(s_i)}}$ 的签名来作应答, 这对攻击者 A 最后的伪造是有直接帮助的. 为了解决这个问题, 我们利用了 Hofheinz 的可编程 hash 函数的性质^[13], 即对构成 H 的生成元 g, h , 假设挑战者 C 知道所有 h 的 $P(s_j)$ 次平方根 (所有的 j), 且知道所有 g 的 $P(s_j)$ 次平方根 (所有 $s_j \neq s_i$). 当攻击者 A 对 M_j ($s_j = s_i = s^*$) 进行签名查询时, 一定有 $H(M_j) \in \langle h \rangle$, 则挑战者 C 可以计算出 $H(M_j)^{(1/2)^{P(s_j)}}$ 作应答. 同时, 要保证 $H(M^*) \notin \langle h \rangle$, 它必须含有非平凡的 g 因子, 这样 C 才可以利用 A 的伪造 () 来解决因子分解问题. 上述解决方法 hash 的条件是满足 $s_j = s_i$ 的下标个数不超过 m (即不存在 $m+1$ 碰撞), 这个概率可以由引理 2.1 计算出来.

下面, 我们通过证明两个引理来证明上述定理.

引理 4.1 如果 A 是一个 I 型攻击者能以 (t, q, ε) 攻破 3 节的签名方案, 则存在一个挑战者 C 能以 (t', ε') 解决因子分解问题, 且 $t \approx t'$, $\varepsilon' \geq \frac{1}{2}(\frac{\delta}{q}(\varepsilon - \frac{q^{m+1}}{2^{m\lambda}}) - \gamma)$.

证明 我们通过攻击者 A 和挑战者 C 之间的交互游戏来证明这个引理. 我们向挑战者 C 给定因子分解问题的一个挑战, 即大整数 N , 目标是输出 N 的因子. 我们只允许攻击者 A 对签名预言进行最多 q 次查询, 并且每次的查询互不相同.

建立: 假如存在 $m+1$ 个下标 $i_1, \dots, i_{m+1}$, 对所有 $j, j' \in \{i_1, \dots, i_{m+1}\}$, 满足 $s_j = s_{j'}$, 则挑战者 C 放弃 (此处放弃的原因见前文我们对 I 型攻击证明的说明), 根据引理 2.1, 放弃的概率是 $\frac{q^{m+1}}{2^{m\lambda}}$. 否则 C 随机选取 q 个 λ -bit 的随机值 $s_1, \dots, s_q$ (: , , hash). 接下来 C 猜测一个下标 $i \in [1, q]$ 满足 $s^* = s_i$, 即 $s^* \in \{s_1, \dots, s_q\}$, 猜中的概率是 $1/q$. 猜测完上述值以后, 挑战者 C 执行如下的过程:

1. 选择伪随机函数的种子 K .

2. 计算出 $c = (\sum_{i=1}^{\lambda} F_K(s_i^*)) \oplus 1$. 构造出 $H_K(x) = F_K(x) \oplus c$.

然后, 令 $E = \{s_1, \dots, s_q\}$, C 任选 $\hat{g}, \hat{h} \in \mathbb{Z}_N^*$, 计算 $g = \hat{g}^{\prod_{t \in E} 2^{P(t)}} \bmod N$, $h = \hat{h}^{\prod_{t \in E} 2^{P(t)}} \bmod N$, 则 $g, h \in Q_N$. 最后 C 运行陷门密钥生成算法 $\text{PHF.TrapGen}(1^k, g, h)$ 获得 hash 密钥 κ 和陷门信息 τ , 我们使用的 PHF 是一个 $(m, 1, \gamma, \delta)$ - 可编程 hash 函数, 因此, PHF.TrapGen 产生的 hash 密钥和 PHF.Gen 产生的 hash 密钥以概率 γ 接近. 最后, C 发送公钥 $\text{PK} = (N, \kappa, K, c)$ 给 A , 并且回答它的签

名查询.

签名查询: 攻击者 A 有选择的对挑战者 C 进行签名查询, 当 A 对消息 M_i 进行查询时, C 首先运行 $\text{PHF.TrapEval}(\tau, M_i)$ 获得 (a_i, b_i) . 如果 $i \in [q]$ 满足 $s_i = s^*$ 时, $a_i \neq 0$, 则 C 放弃 (此处必须 $a_i = 0$, 才能保证 $H(M_i) \in \langle h \rangle$, 见前文对 I 型攻击证明的说明). 因为 PHF 是一个 $(m, 1, \gamma, \delta)$ - 可编程 hash 函数, 所以 C 不放弃的概率为 δ . 令 $E_i = E \setminus \{s_i\}$, C 计算

$$\sigma_i = \hat{g}^{a_i \prod_{t \in E_i} 2^{P(t)}} \cdot \hat{h}^{b_i \prod_{t \in E_i} 2^{P(t)}} = H(M_i)^{\frac{1}{2^{P(s_i)}}} \bmod N,$$

将签名 (M_i, σ_i) 作为应答返回给 A .

输出: 最终攻击者 A 输出伪造签名 (M^*, σ^*, s^*) . 由验证等式且 $P(s^*) = 1$, 则下式成立

$$\sigma^{*2} = (H(M^*)^{\frac{1}{2^{P(s^*)}}})^2 = H(M^*) \bmod N.$$

挑战者 C 运行 $\text{PHF.TrapEval}(\tau, M^*)$ 获得 (a^*, b^*) ($a^* \neq 0$), 然后令 $E^* = E \setminus \{s^*\}$, 计算

$$b = \hat{g}^{a^* \prod_{t \in E^*} 2^{P(t)}} \cdot \hat{h}^{b^* \prod_{t \in E^*} 2^{P(t)}} \bmod N,$$

则 $\sigma^{*2} = b^2 \bmod N$. 如果 $\sigma^* \neq \pm b$, 则挑战者 C 可通过计算 $\text{GCD}(\sigma^* - b, N)$ 或者 $\text{GCD}(\sigma^* + b, N)$ 输出 N 的因子. 因为 $\hat{g}, \hat{h}$ 是随机选取的, 所以 σ^* 和 $\pm b$ 相异的概率为 $1/2$.

假如攻击者 A 能以概率 ε 赢得上面的交互游戏, 挑战者 C 在游戏过程不放弃且猜测正确, 则我们可以推出 C 可利用 A 至少以概率 $\frac{1}{2}(\frac{\delta}{q}(\varepsilon - \frac{q^{m+1}}{2^{m\lambda}}) - \gamma)$ 解决因子分解问题, 输出 N 的因子.

引理 4.2 如果 A 是一个 II 型攻击者能以 (t, q, ε) 攻破 3 节的签名方案, 则存在一个挑战者 C 能以 (t', ε') 解决因子分解问题, 且 $t \approx t'$, $\varepsilon' \geq \frac{\varepsilon - \gamma}{2(q+1)\lambda}$.

证明 我们依然通过攻击者 A 和挑战者 C 之间的交互游戏来证明这个引理. 我们向挑战者 C 给定因子分解问题的一个挑战, 即大整数 N , 目标是输出 N 的因子. II 型攻击者最终返回的伪造签名 (M^*, σ^*, s^*) , $s^* \notin \{s_1, \dots, s_q\}$.

建立: C 随机选取 q 个 λ -bit 的随机值 $s_1, \dots, s_q$. 接下来 C 猜测一个下标 $i \in [1, q]$ 满足 s^* 和 s_i 有最长的共同前缀, 设共同前缀的长度为 $j-1$, 令 $\psi = s_{i|j-1} || x$ (x 是 s_i 第 j 个比特的相反码). 如果 $q = 0$, 则 ψ 随机选择 0 或者 1. 如果 ψ 不是 s^* 的前缀或者 ψ 是 $\{s_1, \dots, s_q\}$ 中某个值的前缀, 则 C 放弃. 此时 C 猜中的概率是 $\frac{1}{(q+1)\lambda}$. 接下来挑战者 C 执行如下的过程:

1. 选择伪随机函数的种子 K .
2. 计算出 $c = F_K(\psi) \oplus 1$. 构造出 $H_K(x) = F_K(x) \oplus c$.

然后, 令 $E = \{s_1, \dots, s_q\}$, C 任选 $\hat{g}, \hat{h} \in \mathbb{Z}_N^*$, 计算

$$g = \hat{g}^{2 \prod_{t \in E} 2^{P(t)}} \bmod N, \quad h = \hat{h}^{2 \prod_{t \in E} 2^{P(t)}} \bmod N,$$

则 $g, h \in Q_N$. 最后 C 运行陷门密钥生成算法 $\text{PHF.TrapGen}(1^k, g, h)$ 获得 hash 密钥 κ 和陷门信息 τ . 因为 PHF 是一个 $(m, 1, \gamma, \delta)$ - 可编程 hash 函数, 因此, PHF.TrapGen 产生的 hash 密钥和 PHF.Gen 产生的 hash 密钥以概率 γ 接近. 最后 C 发送公钥 $\text{PK} = (N, \kappa, K, c)$ 给 A , 并且回答它的签名查询.

签名查询: 攻击者 A 有选择的对挑战者 C 进行签名查询, 当 A 对消息 M_i 进行查询时, C 首先运行 $\text{PHF.TrapEval}(\tau, M_i)$ 获得 (a_i, b_i) . 令 $E_i = E \setminus \{s_i\}$, C 计算

$$\sigma_i = \hat{g}^{2a_i \prod_{t \in E_i} 2^{P(t)}} \cdot \hat{h}^{2b_i \prod_{t \in E_i} 2^{P(t)}} = H(M_i)^{\frac{1}{2^{P(s_i)}}} \bmod N,$$

将签名 (M_i, σ_i) 作为应答返回给 A .

输出: 最终攻击者 A 输出伪造签名 (M^*, σ^*, s^*) . 挑战者 C 计算 $z = \sum_{\{i \in [\lambda] | s_i^* \neq \psi\}} H_K(s_i^*)$, 再计算 $\omega = \sigma^{*2^z}$, 由验证等式且 $P(\psi) = 1$, 下式成立

$$\omega^2 = (\sigma^{*2^z})^2 = H(M^*) \bmod N.$$

挑战者 C 运行 $\text{PHF.TrapEval}(\tau, M^*)$ 获得 (a^*, b^*) , 计算

$$y = \hat{g}^{a^* \prod_{t \in E} 2^{P(t)}} \cdot \hat{h}^{b^* \prod_{t \in E} 2^{P(t)}} \bmod N,$$

则 $\sigma^{*2} = y^2 \bmod N$. 如果 $\sigma^* \neq \pm y$, 则挑战者 C 可通过计算 $\text{GCD}(\sigma^* - y, N)$ 或者 $\text{GCD}(\sigma^* + y, N)$ 输出 N 的因子. 因为 $\hat{g}, \hat{h}$ 是随机选取的, 所以 σ^* 和 $\pm y$ 相异的概率为 $1/2$.

假如攻击者 A 能以概率 ε 赢得上面的交互游戏, 综合上面的分析, 我们可以推出 C 可利用 A 至少以概率 $\frac{\varepsilon - \gamma}{2(q+1)\lambda}$ 解决因子分解问题, 输出 N 的因子.

综合引理 4.1 和 4.2 可知定理 4.1 成立.

注意: 在 Hohenberger 等^[12]的 RSA 签名方案中, 没有随机值, 而将 $P(M)$ 放在签名的指数部分. 但是在安全性证明中, 待查询签名的消息 M 是不能由挑战者 C 事先选好的, 如果由 C 强行选好, 则只能证明方案是弱安全的^[12]. 由弱安全的签名方案转化为标准安全的签名方案, 需要嵌入变色龙 hash 函数^[12], 这降低了签名的效率. 相比之下, 在我们的方案中, $P(s)$ 位于签名的指数部分, 而随机值是在签名过程中产生的. 因此在安全性证明中 (引理 4.1 和 4.2), 应答查询所用的随机值可由挑战者 C 事先选好, 最后直接完成标准安全性证明, 而不需要嵌入变色龙 hash 函数.

5 结束语

可编程 hash 函数的特点是可以模拟随机预言的某些性质, 对签名的构造非常有用. 本文利用 $(m, 1, \gamma, \delta)$ - 可编程 hash 函数设计了一个高效的短签名方案. 它的优势在于, 首先不需要嵌入变色龙 hash 函数便可实现标准模型下的可证安全; 其次是签名只包含一个 $\mathbb{Z}_N$ 上的元素和一个小整数, 长度很短; 最重要的是签名过程不再需要产生奇素数, 提高了计算效率, 也回答了 Hofheinz 等的公开问题. 由于本文方案的上述有优点, 它特别适合无线传感网、物联网等资源受限的场合.

参考文献

- 1 Diffie Z, Hellman M E. New directions in cryptography. IEEE Trans Inf Theory, 1976, 22: 644–654
- 2 Rivest R, Shamir A, Adleman L. A method for obtaining digital signatures and public key cryptosystem. Commun ACM, 1988, 21: 120–126
- 3 Bellare M, Rogaway P. The exact security of digital signatures: How to sign with RSA and Rabin. In: Maurer U M, ed. Advances in Cryptology-EUROCRYPT'96, Saragossa, 1996. LNCS 1070. Berlin: Springer. 399–416
- 4 Boneh D, Lynn B, Shacham H. Short signatures from the Weil pairing. In: Boyd C, ed. Advances in Cryptology - ASIACRYPT 2001, Gold Coast, 2001. LNCS 2248. Berlin: Springer. 514–532
- 5 Bellare M, Ristenpart T. Simulation without the artificial abort: Simplified proof and improved concrete security for Waters' IBE scheme. In: Joux A, ed. EUROCRYPT, 2009. LNCS 5479. Berlin: Springer. 407–424
- 6 Canetti R, Goldreich O, Halevi S. The random oracle methodology, revisited (preliminary version). In: 30th Annual ACM Symposium on Theory of Computing, Dallas, 1998. 209–218

- 7 Fischlin M. The Cramer-Shoup strong-RSA signature scheme revisited. In: Desmedt Y, ed. 6th International Workshop on Theory and Practice in Public Key Cryptography, Miami, 2003. LNCS 2567. Berlin: Springer. 116–129
- 8 Cramer R, Shoup V. Signature schemes based on the strong RSA assumption. In: 6th Conference on Computer and Communications Security (ACM CCS), Singapore, 1999. 46–51
- 9 Hofheinz D, Kiltz E. Programmable hash functions and their applications. In: Wagner D, ed. Advances in Cryptology (CRYPTO), Santa Barbara, 2008. LNCS 5157. Berlin: Springer. 21–38
- 10 Hofheinz D, Kiltz E. Programmable hash functions and their applications. J Cryptol, 2011
- 11 Boneh D, Boyen X. Short signatures without random oracles. In: Cachin C, Camenisch J, eds. Advances in Cryptology (EUROCRYPT), Interlaken, 2004. LNCS 3027. Berlin: Springer. 56–73
- 12 Hohenberger S, Waters B. Short and stateless signatures from the RSA assumption. In: CRYPTO, Santa Barbara, 2009. LNCS 5677. Berlin: Springer. 654–670
- 13 Hofheinz D, Jager T, Kiltz E. Short signatures from weaker assumptions. Cryptology ePrint Archive: Report 2011/296, <http://eprint.iacr.org/2011/296>
- 14 Shoup V. A Computational Introduction to Number Theory and Algebra. Singapore: Cambridge University Press, 2005. 534

Short signature based on programmable hash functions

WANG ZhiWei^{1,2,3,4}

1 College of Computers, Nanjing University of Posts and Telecommunications, Nanjing 210003, China;

2 State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China;

3 Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210003, China;

4 Key Laboratory of Broadband Wireless Communication and Sensor Network Technology, Ministry of Education Jiangsu Province, Nanjing 210003, China

E-mail: zhwwang@njupt.edu.cn

Abstract A short signature, which is a type of digital signature, has an advantage in terms of its length. Short signatures are particularly suitable for scenarios in which communication bandwidth is constrained. Most of the existing short signature schemes have been proven secure under a random oracle, which is considered to be too idealistic. There is no hash function that can simulate a random oracle. A few of the provable secure short signature schemes in the standard model are considered to be inefficient or subject to strong hard assumptions, where the attacker is required to solve a chosen instance given a certain number of solved instances. Programmable hash functions (PHFs) can mimic some of the programmability properties of random oracles. Thus, PHFs can be plugged into the generic construction of signatures to yield short signatures in the standard model. We propose a short signature scheme based on a factoring assumption that uses programmable hash functions. The advantages are that: 1) the short signature only involves one element in a group and one small integer; 2) the scheme is efficient in terms of computational cost and does not require the generation of primes at signing; and 3) it is provably secure under the standard model without Chameleon hashes. This scheme is especially suitable for resource constrained applications such as wireless sensor networks and the Internet of things.

Keywords short signature, cryptography, factoring assumption, programmable hash functions, provable security

WANG ZhiWei was born in 1976. He received his Ph.D. degree in cryptography from the Beijing University of Posts and Telecommunications, Beijing in 2009. Currently, he is an Associate Professor in the College of Computers at Nanjing University of Posts and Telecommunications. His research interests include digital signatures, provable security, cryptographic protocols, and network and cloud security. Dr.

Wang was a TPC member for MobiPST 2011, 2012, and 2013, ACSA-Summer 2012, ICITIS 2012, and MIST 2012.