中国科学: 信息科学 2010 年 第 40 卷 第 6 期: 786～795
www.scichina.com info.scichina.com
《中国科学》杂志社
SCIENCE CHINA PRESS
论 文

公平交换签名方案

刘景伟[①②*], 孙蓉[①], 郭庆燮[②]

① 计算机网络与信息安全教育部重点实验室, 西安电子科技大学, 西安 710071, 中国
② UWB 无线通信研究中心, 仁荷大学, 仁川 402751, 韩国
* 通信作者. E-mail: j_w_liu@hotmail.com

收稿日期: 2008–07–10; 接受日期: 2009–07–29
韩国 NIPA 项目 (批准号: NIPA-2009-C1090-0902-0019)、韩国高等教育财团国际学术交流项目 (2009-2010) 和中国高等学校学科创新引智计划 (批准号: B08038) 资助

摘要 本文提出一类新的公平交换签名方案 (fair exchange signature scheme, FESS), 该方案可以使参与双方以一种公平的方式交换数字签名. 因为 FESS 所给出的是一种构造公平交换签名方案基本模型, 所以它可以基于大多数现在已有的签名方案来实现, 本文以 Schnorr 签名为基础给出了 FESS 的一种实现方案. 可以把 FESS 看作是 EUROCRYPT2004 上发表的同时生效签名 (concurrent signature) 的一种有趣的扩展变形. 同时生效签名只是基于环签名来构造的, 而 FESS 能够基于大部分普通数字签名来实现. 在 FESS 中, 参与双方分别签署两个能够被对方验证但是却未被激活的数字签名, 直到承诺信息 — keystone 被公开以后, 双方的数字签名才能够被激活, 并且同时生效. 一旦 keystone 被公开以后, 任何人都可以验证两个签名的合法性. FESS 的一个显著特点是两个参与者能够通过一个安全承诺 — keystone 来同时交换各自的数字签名, 而不涉及任何可信第三方 (TTP), 而且该方案的执行效率要高于同时生效签名. FESS 的提出为构造公平电子支付协议以及公平电子合同签署协议提供了非常有效的密码工具.

关键词
FESS
同时生效签名
Schnorr 签名
公平交换
电子商务

1 引言

随着 Internet 这样的开放式网络的迅速发展, 网络已经给我们开展大规模电子商务、电子政务等应用提供了坚实的基础. 而其中往往涉及到网络上互不信任的两方进行数字信息的交换, 例如: 电子商务支付协议、电子合同的签署, 以及认证电子邮件的发送等等. 由于电子商务的迅速发展, 这些电子信息的公平交换变得日益重要. 一个数字交换问题是否公平, 要看在交换结束时参与双方是否都收到了各自期望的信息或者都没有收到. 一般来说, 数字信息的交换通常不得不在开放的互联网上由互相不信任的双方进行. 即使在一次公平交易后, 双方也有可能对交易的内容有所争议, 在这种情况下, 对交易过程中证据的收集就显得尤为重要, 这可以确保在未来可能发生的纠纷能够被解决.

引用格式: 刘景伟, 孙蓉, 郭庆燮. 公平交换签名方案. 中国科学: 信息科学, 2010, 40: 786–795

1.1 研究现状

近些年来, 研究人员在各种文献上提出了许多关于公平交换问题的解决方案. 这些方案无外乎 3 个方面:

一、早期的许多公平交换问题的解决方案主要是渐进交换协议 [1~7], 其中参与双方不得不进行大量而且繁琐的交互步骤以达到交换数字信息的目的. 然而, 这些解决方法仍不能保证完整的公平性, 因为在协议结束的时候, 一方常常比另一方拥有 1 bit 的优势. 在文献 [3] 中, 作者引入了时间承诺这个概念. 时间承诺是一种承诺机制, 它有一个可选的强制公开措施来保证接收者能够不用承诺者的帮助而 (经过一定的努力) 恢复出承诺值. 但是这种方法只限于 Rabin 和 RSA 这两类特殊的签名方案. 在文献 [7] 中, 作者给出了如何达到时效公平交换数字签名的标准方法, 该方法的构造遵循逐步公开的规范, 它基于一种新的叫做镜像时间的时间结构. 然而它的长度却又导致了另外一个很明显的问题, 就是为了保证可能的顺序有一个足够大的期限, 导致了交互过程是不可观测的.

二、最近, 许多关于公平交换协议的研究主要是集中在利用在线或者离线可信第三方 (TTP)[8~21] 上, 在这一类协议中 TTP 要么一直保持在线 (on-line) 或者在发生纠纷的时候出面解决 (off-line). 因此, 不管是 on-line 还是 off-line 的 TTP 都很容易引起瓶颈问题, 至少是执行效率上的低下. 尽管后来文献 [8, 12, 20] 针对这一问题作了进一步的研究, 但仍没有很好的解决方法. 在文献 [8] 中作者介绍了一种新的协议, 该协议允许两个参与者在互联网上以一种公平的方式交换数字签名. 协议中用到了可信第三方 TTP, 但却是 “乐观的”, 因为第三方只在发生一方企图欺骗和简化冲突的情况下出现. 该协议的一个重要特点是任意一方总是能够保证协议有时限的公平的结束, 而不需要另一方的帮助.

三、关于公平交换的最新研究方向是在协议的过程中舍弃可信第三方 (TTP), 直接由参与交换的双方通过使用特殊的数字签名来交换数字信息. 通过这种方法可以设计出许多新的公平交换协议, 而且这种方法要比以前的方法有效的多. 同时生效签名 (concurrent signature)[22] 这一概念是由 Chen 等人在 EUROCRYPT 2004 上首先提出来的. 在这种签名方案里面, 两名参与者生成并交换两个模糊的数学签名, 当额外的信息 — keystone 由一方公布后, 签名的模糊性消失, 这一性质是利用了环签名的模糊性 [23,24]. 该签名方案最特殊的地方是在 keystone 公布之前, 两个签名究竟分别对应着哪一方的身份是模糊的, 比如这两个签名有可能是有双方共同签署的, 也有可能只是由一方签署的. 然而当 keystone 被大家知道以后, 两个签名将同时绑定它们真实的签名者的身份, 此时, 任何第三方都可以验证究竟是谁签署了哪一个签名. 同时生效签名可以用来构造只有两方通过交互来交换数字信息的公平交换协议, 而不涉及可信第三方 (TTP). 但是这种良好性质的实现是需要一定的条件的, 就是 keystone 由发起方控制着, 因此他有额外的权力可以决定什么时间公开 keystone 或者就根本不公开. 在 ICICS’04 上, Susilo, Mu 和 Zhang 等 [25] 进一步提出了完美同时生效签名来加强同时生效签名的模糊性, 即使两个签名者的身份都已经被公开, 知道是由他们签署了两个模糊签名, 任何第三方仍然不能推导出是谁签署了哪一个签名, 这一点与 Chen 等的方案是不同的. 但是很快 Wang 等就在文献 [26] 中指出, Susilo 等提出的两种完美同时生效签名并不是真正的同时生效签名, 并且又提出了一种有效的方法来避免这种攻击.

1.2 主要工作

先前的对于公平交换问题的大部分研究工作并不能满足在开放网络上的实际应用, 因为大多数的公平交换应用不仅仅要求要有一定的安全性而且要有效率. 所以我们认为一个理想的公平交换解决方案应该同时满足安全性和高效性. 按照这种想法, 本文提出一类新的公平交换签名方案 (fair exchange

signature scheme, FESS), 这种方案可以保证两个参与者在公开的计算机通信网络上 (例如, 互联网 Internet) 以一种公平的方式交换各自的数字签名, 因为每个参与者要么都能够同时得到对方的合法数字签名, 或者都得不到. 在此方案中, 每一个分别由两个参与者各自签署的未被激活的签名都能够被其他任何第三方验证. 但是, 它却并不能马上生效, 必须等待协议的发起者公布一条额外的信息 keystone, 一旦 keystone 被公开, 两个签名才能够同时被激活并且同时生效. 这种签名方案可以根据不同的执行效率和安全要求而选择不同的实现算法来应用于不同的环境之中. 后面也将介绍如何构造一个不涉及可信第三方 (TTP) 的 FESS 的具体实现. 因此, 它对设计公平交换方案提供了一种基本而且有效的途径. 当然, 也可以把它应用到其他一些可能的应用环境中. FESS 方案的一个重要特点是两方直接参与并完成数字签名的公平交换, 而不涉及任何可信第三方 (TTP). 虽然 FESS 并没有克服同时生效签名的弱点, 就是由签名的发起者控制 keystone, 但是它确实比同时生效签名有着更高的执行效率.

本文的主要工作如下:

1) 提出了公平交换数字签名方案 — FESS 的定义;

2) 给出了一个没有可信第三方 (TTP) 的 FESS 的具体实例;

3) 给出了 FESS 在随机预言机模型下的安全性证明.

2 基本定义

本节介绍 FESS 的一些基本定义. 首先介绍新方案中所涉及到的一些参数.

- 明文消息空间 $\mathcal{M}$: 一个字符上的消息流集合;
- keystone 消息空间 $\mathcal{K}_{\rm ks}$: 一个字符上的集合;
- keystone 的承诺空间 $\mathcal{K}$: 一个可能的 keystone 的映射集合;
- 数字签名空间 $\mathcal{S}$: 一个可能的数字签名消息集合;
- 私钥空间 $\mathcal{X}$: 一个可能的创建签名的私钥集合;
- 公钥空间 $\mathcal{Y}$: 一个可能的验证签名的公钥集合.

定义 1 一个完整的 FESS 方案包括 5 个部分 (Parameter Setup, KGen, Sign, SVerify, KVerify):

- 一个有效的概率算法 Parameter Setup:

$$k \to \langle \{x_i\}, \{y_i\}, \{\mathcal{M}, \mathcal{K}_{\rm ks}, \mathcal{K}, \mathcal{S}\text{的描述}\}\rangle,$$

这里 k 是一个安全参数, $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$.

- 一个有效的单向函数 KGen:

$$\mathcal{K}_{\rm ks} \to \mathcal{K},$$

使用安全参数 keystone $\in \mathcal{K}_{\rm ks}$ 作为输入用来生成 keystone 的一个承诺值 $k \in \mathcal{K}$. 安全 Hash 函数可以用来作为好的 KGen 算法.

- 一个高效的概率签名算法 Sign:

$$\mathcal{M} \times \mathcal{K} \times \mathcal{X} \to \mathcal{S},$$

对于任何消息 $m \in \mathcal{M}$, keystone 承诺 $k \in \mathcal{K}$ 和私钥 $x \in \mathcal{X}$, 我们规定 $s \leftarrow {\rm Sign}_x(m, k)$, 这里 $s \in \mathcal{S}$.

- 一个有效的签名验证算法 SVerify:

$$\mathcal{M} \times \mathcal{K} \times \mathcal{S} \times \mathcal{Y} \to \{{\rm True}, {\rm False}\},$$

对于任何 $m \in \mathcal{M}$, $k \in \mathcal{K}$, $y \in \mathcal{Y}$, 都满足

$$\mathrm{SVerify}_y(m,k,s) = \text{True 或 False}. \tag{1}$$

• 一个有效的 keystone 验证算法 KVerify:

$$\mathcal{M} \times \mathcal{K} \times \mathcal{S} \times \mathcal{Y} \times \mathcal{K}_{ks} \to \{\mathrm{True}, \mathrm{False}\},$$

对于任何 $m \in \mathcal{M}$, $k \in \mathcal{K}$, $y \in \mathcal{Y}$, keystone $\in \mathcal{K}_{\mathrm{ks}}$, 都满足

$$\mathrm{KVerify}_y(m,k,s,\mathrm{keystone}) = \begin{cases} \text{True, 如果 } \mathrm{SVerify}_y(m,k,s) = \text{True 并且 } k = \mathrm{KGen(keystone)}, \\ \text{False, 其他情况.} \end{cases} \tag{2}$$

3 基本模型

3.1 公平交换签名协议

一般情况下, 大多数公平交换方案常常要涉及到在线的或者离线的可信第三方. 但是 FESS 的一个显著优点是只需要参与双方就可以完成公平交换, 而不需要任何可信第三方的参与. 不失一般意义, 假设参与双方为 Alice (发起签名者) 和 Bob (响应签名者). 发起签名者 Alice 首先产生一条随机安全信息 — keystone, 然后使用她自己的私钥和以 keystone 作为安全单向函数输入而产生的 keystone 承诺来签署一条消息, 并且把消息和签名发送给响应签名者 Bob. Bob 用他自己的私钥和同一个 keystone 承诺来签署另一条消息以回应 Alice 的消息.

根据定义 1, FESS 的详细执行步骤如下描述:

Alice 和 Bob 首先选择一套有效的签名方案以及相应的参数. 假设 $x_{\mathrm{A}}, x_{\mathrm{B}} \in \mathcal{X}$ 分别表示 Alice 和 Bob 的私钥, $y_{\mathrm{A}}, y_{\mathrm{B}} \in \mathcal{Y}$ 分别表示与两个参与者相对应的公钥.

1) Alice 随机选择一个 keystone $\in \mathcal{K}_{\mathrm{ks}}$ 并且计算 $k = \mathrm{KGen(keystone)}$, 这里 $k \in \mathcal{K}$. 然后她使用 k 和她自己的私钥 x_{A} 来的对一条 m_{A} 消息进行签名 $s_{\mathrm{A}} = \mathrm{Sign}_{x_{\mathrm{A}}}(m_{\mathrm{A}}, k)$, 消息 m_{A} 可以事先与 Bob 商订. 然后将可以验证的签名消息 $\sigma = \langle m_{\mathrm{A}}, k, s_{\mathrm{A}} \rangle$ 发送给 Bob.

2) 在接收到 Alice 的可以验证的签名消息 σ_{A} 后, Bob 首先使用在前面定义的算法 SVerify 验证这条签名消息 σ_{A}. 如果 $\mathrm{SVerify}_{y_{\mathrm{A}}}(\sigma_{\mathrm{A}}) = \text{True}$, Bob 则选择一条消息 m_{B} 并且使用 k 和他自己的私钥 x_{B} 来对消息 m_{B} 进行签名 $s_{\mathrm{B}} = \mathrm{Sign}_{x_{\mathrm{B}}}(m_{\mathrm{B}}, k)$, 这里消息 m_{B} 同样可以事先与 Alice 商订. Bob 把这条可以验证的签名消息 $\sigma_{\mathrm{B}} = \langle m_{\mathrm{B}}, k, s_{\mathrm{B}} \rangle$ 返还给 Alice. 注意: 这里 Bob 使用与 Alice 相同的 keystone 承诺值 k. 相反地, 如果 $\mathrm{SVerify}_{y_{\mathrm{A}}}(\sigma_{\mathrm{A}}) = \text{False}$, Bob 则终止签名步骤.

3) 在接收到 Bob 的可以验证的签名消息 σ_{B} 后, Alice 同样使用算法 SVerify 来验证这条签名消息 σ_{B}. 如果 $\mathrm{SVerify}_{y_{\mathrm{B}}}(\sigma_B) = \text{True}$, 公布 keystone 来同时激活签名 σ_{B} 和 σ_{A}, 进而两个签名同时生效. 如果 $\mathrm{SVerify}_{y_{\mathrm{B}}}(\sigma_{\mathrm{B}}) = \text{False}$, Alice 则终止签名步骤.

4) 任何人都可以验证 $\mathrm{KVerify}_{y_{\mathrm{A}}}(\sigma_{\mathrm{A}}, \mathrm{keystone}) = \text{True}$ 和 $\mathrm{KVerify}_{y_{\mathrm{B}}}(\sigma_{\mathrm{B}}, \mathrm{keystone}) = \text{True}$ 是否成立.

这里, 需要指出的是 FESS 是通过未激活签名的沉睡性来保证公平的, 这一点与同时生效签名 [22] 中所用到的模糊性不同. 作为一个有用的密码工具, 此方案为构造公平交换和合同签署协议提供了一种有效的方法. 在下一部分, 将给出一个具体的实现例子.

3.2 FESS 的攻击模型

作为一个安全签名方案, 首先应该具有能够在适应性选择消息攻击下抵抗存在性伪造数字签名的良好的安全性质. 在这种安全模型下 [27,28], 假如一个敌手能够输出一个有效的消息和签名对就算赢得游戏. 在这个过程中, 他可以要求签名者签署除了输出签名外的任何消息. 在这里将引入 FESS 的一个攻击模型, 跟文献 [28] 中介绍的很类似. FESS 中共包含 5 个算法: Parameter Setup, KGen, Sign, SVerify, KVerify. 我们称 FESS 在适应性选择消息攻击下能够抵抗存在性伪造数字签名, 如果在下面的游戏中不存在多项式时间算法 $\mathcal{A}$ 能够借助一个挑战者 $\mathcal{S}$ 拥有一个不可忽略的概率优势赢得游戏:

1) $\mathcal{S}$ 首先运行 Parameter Setup 算法然后将公共系统参数送给 $\mathcal{A}$.

2) $\mathcal{A}$ 可以要求回答下列问题:

i) Hash Query: $\mathcal{S}$ 利用 Hash 函数计算所要求输入的杂凑值并且把这个值送给 $\mathcal{A}$.

ii) KGen Query: $\mathcal{A}$ 可以要求 $\mathcal{S}$ 选择一个 keystone $\in \mathcal{K}_{\rm ks}$ 并返回 keystone 承诺 $k = {\rm KGen(keystone)}$.

iii) KReveal Query: $\mathcal{A}$ 可以要求得到先前使用 KGen Query 得到的任何 keystone 承诺 $k \in \mathcal{K}$ 所对应的 keystone.

iv) Sign Query: 给定一条消息 $m \in \mathcal{M}$ 和一个 $k \in \mathcal{K}$, $\mathcal{S}$ 使用签名算法 Sign 得到并回复一个签名 s.

3) $\mathcal{A}$ 输出 $\langle m, k, s\rangle$, 这里 m 是一条消息, k 是 keystone 承诺, s 是对应消息的签名, 而且 $\langle m, k\rangle$ 并没有在 Sign 的提问序列中出现过而且 k 是先前 KGen Query 的一个输出, 是先前 KReveal Query 的一个输入. $\mathcal{A}$ 赢得游戏, 如果 s 是 $\langle m, k\rangle$ 的一个合法签名.

后面使用这个攻击模型将 FESS 的安全性归约到了离散对数问题的困难性.

4 基于 Schnorr 签名的 FESS 方案

在这一部分, 我们给出了一个基于 Schnorr 签名的 FESS 具体实现方案. 首先给出系统参数.

参数设定:

• 系统参数: 设 p 和 q 是两个大素数, 设 p 和 q 是两个大素数, 并且满足 $q|p-1$. 符号 g 表示 $\mathbb{Z}_p$ 中的一个 q 阶元素.

• Alice: Alice 拥有一个 Schnorr 签名的公私钥对 $(x_{\rm A}, y_{\rm A})$, 这里 $x_{\rm A}$ 是 Alice 的私钥, $y_{\rm A}$ 是她的公钥, 并且满足 $y_{\rm A} = g^{-x_{\rm A}} \bmod p$.

• Bob: Bob 同样拥有一个 Schnorr 签名的公私钥对 $(x_{\rm B}, y_{\rm B})$, 这里 $x_{\rm B}$ 是 Bob 的私钥, $y_{\rm B}$ 是他的公钥, 并且满足 $y_{\rm B} = g^{-x_{\rm B}} \bmod p$.

1) Alice 首先选择一个 keystone $=\langle {\rm ID_{AB}}\rangle$, 计算 $k = G({\rm ID_{AB}})$, 这里 ${\rm ID_{AB}}$ 是关于 Alice 和 Bob 身份的一些随机信息, G 是一个杂凑函数. Alice 计算生成她的签名消息 $s_{\rm A} = {\rm Sign}_{x_{\rm A}}(m_{\rm A}, k) = \langle r_{\rm A}, e_{\rm A}, c_{\rm A}\rangle$, 这里 $r_{\rm A} = g^{k_{\rm A}} \bmod p$, $e_{\rm A} = H(m_{\rm A}, k, r_{\rm A})$, $c_{\rm A} = k_{\rm A} + e_{\rm A}x_{\rm A} \bmod q$, 其中 H 是一个杂凑函数. Alice 把生成的可以验证的签名消息 $\sigma_{\rm A} = \langle m_{\rm A}, k, s_{\rm A}\rangle$ 发送给 Bob.

2) Bob 使用算法 SVerify 验证 $\sigma_{\rm A}$. 如果 $e_{\rm A} = H(m_{\rm A}, k, g^{c_{\rm A}} y_{\rm A}^{e_{\rm A}} \bmod p)$, Bob 计算 $s_{\rm B} = {\rm Sign}_{x_{\rm B}}(m_{\rm B}, k) = \langle r_{\rm B}, e_{\rm B}, c_{\rm B}\rangle$ 并发送给 Alice. 否则的话 Bob 什么都不做, 终止会话.

3) Alice 验证 $\sigma_{\rm B}$. 如果 $e_{\rm B} = H(m_{\rm B}, k, g^{c_{\rm B}} y_{\rm B}^{e_{\rm B}} \bmod p)$, Alice 公布 keystone. 否则的话 Alice 什么都不做, 终止会话.

4) 任何人都能够通过验证下面两个等式来验证签名 $\sigma_{\rm A}$ (或者 $\sigma_{\rm B}$) 是否有效, $k = G({\rm ID_{AB}})$, ${\rm SVerify}_{y_{\rm A}}(\sigma_{\rm A}) = {\rm True}$, (或者 ${\rm SVerify}_{y_{\rm B}}(\sigma_{\rm B}) = {\rm True}$).

从上面的具体实现, 可以得出结论, FESS 是一个基本的概念并且有可以基于大部分已有的数字签名方案来得到不同的实现. 也可以把它看作是 EUROCRYPT 2004 上提出的基于环签名实现的同时生效签名 [22] 的一种有趣的变形. 因为 FESS 可以通过更为简单高效的数字签名方案来实现, 所以它比同时生效签名有着更为高效的执行效率. 在下一部分将给出 FESS 和同时生效签名的效率比较.

5 FESS 的安全性与效率分析

5.1 安全性分析

这一部分将讨论 FESS 在随机预言机模型 [29] 下的安全性.

引理 1 (完备性) 公平交换签名方案 FESS 满足完备性.

证明 如果 $s = \mathrm{Sign}_x(m,k) = \langle r,e,c\rangle$, $r = g^{k'} \bmod p$, $e = H(m,k,r)$, $c = k' + ex \bmod q$, 那么 $e = H(m,k,g^c y^e \bmod p)$, $\Leftrightarrow$ $\mathrm{SVerify}_y(m,k,s) = \mathrm{True}$. 更进一步, 如果 $\mathrm{SVerify}_y(m,k,s) = \mathrm{True}$, $k = G(\mathrm{keystone})$, 那么 $\mathrm{KVerify}_y(m,k,s,\mathrm{keystone}) = \mathrm{True}$

引理 2 (不可伪造性) 在随机预言机模型中, FESS 在适应性选择消息攻击下是不可伪造的.

证明 本证明方法参考了 Pointcheval 和 Stern 在文献 [27, 28] 中利用了分叉引理对数字签名方案不可伪造性的证明方法. 假设 G 和 H 随机预言机, $\mathcal{A}$ 是一个概率多项式时间图灵机, 它的输入只是一些公共参数. 假定 $\mathcal{A}$ 可以向随机预言机 G 询问 Q_G 个问题, 向随机预言机 H 询问 Q_H 个问题, 向随机预言机 KGen 询问 Q_K 个问题, 向签名预言机 Sign 询问 R 个问题. 在时间界限 T 内, $\mathcal{A}$ 以一个不可忽略的概率 $\varepsilon \geqslant 10Q_G(R+1)(R+Q_H)/2^q$(这里 q 是一个安全参数) 生成一个有效的签名 $\langle m,k,\langle r,e,c\rangle\rangle$.

模拟仿真 $\mathcal{S}$ 把公共参数 $\langle g,p,q\rangle$ 和 $y = g^{-x} \bmod p$ 给 $\mathcal{A}$. $\mathcal{S}$ 试图通过模拟所有的预言机来得到私钥 x 来模拟挑战者. $\mathcal{A}$ 可以询问下列问题:

G-Queries $\mathcal{A}$ 可以在任意时刻向随机预言机 G 提问问题. $\mathcal{S}$ 通过将二元组 $\langle m_i,k_i\rangle$ 保存到列表 G-List 来模拟随机预言机. 当预言机的出入问题为 $m \in \{0,1\}^*$ 时, $\mathcal{S}$ 做如下回答:

1) 如果输入问题 m 已经在列表 G-List 的二元组 $\langle m,k_i\rangle$ 中, 那么 $\mathcal{S}$ 输出 k_i.

2) 否则的话, $\mathcal{S}$ 随机选取并输出一个 $k \in \mathcal{K}$, 然后将 $\langle m,k\rangle$ 添加到 G-List 中.

H-Queries $\mathcal{A}$ 可以在任意时刻向随机预言机 H 提问问题. $\mathcal{S}$ 通过保存二元组 $\langle \Sigma_i,e_i\rangle$ 的一个列表 H-List 来模拟随机预言机 H, 这里 Σ_i 是一个三元组 $\langle m_i,k_i,r_i\rangle$. 当随机预言机的输入要求为 Σ 时, $\mathcal{S}$ 按如下步骤进行回应:

1) 如果问题 Σ 已经存在于列表 H-List 中三元组 $\langle \Sigma,e_i\rangle$ 中, 那么 $\mathcal{S}$ 输出 e_i;

2) 否则的话, $\mathcal{S}$ 随即选取并输出一个 $e \in \mathbb{Z}_q$, 然后将 $\langle \Sigma,e\rangle$ 添加到列表 H-List 中.

KGen-Queries $\mathcal{S}$ 保存一个二元组 $\langle \mathrm{keystone},k\rangle$ 的列表 K-List. $\mathcal{A}$ 可要求 $\mathcal{S}$ 选择一个 $\mathrm{keystone} \in \mathcal{K}_{\mathrm{ks}}$ 并且返回它的承诺值 $k = G(\mathrm{keystone})$. 这时, $\mathcal{S}$ 随机选取一个 $\mathrm{keystone} \in \mathcal{K}_{ks}$ 并计算 $k = G(\mathrm{keystone})$. $\mathcal{S}$ 输出 k 并且将 $\langle \mathrm{keystone},k\rangle$ 添加到列表 K-List. 实际上, 列表 K-List 是列表 G-List 的一个子列表, 但是不同的是列表 K-List 的结果可以被要求用来回答 KReveal-Queries 的问题.

KReveal-Queries $\mathcal{A}$ 可以要求得到任何先前由 KGen-Queries 生成的 keystone 承诺值 $k \in \mathcal{K}$ 所对应的 keystone 的值. 如果在列表 K-List 中存在一个二元组 $\langle \mathrm{keystone},k\rangle$, 那么 $\mathcal{S}$ 就返回该 keystone 的值, 否则话就输出无效.

Sign-Queries 这里由 $\mathcal{S}$ 来模拟签名预言机, 它首先接收 $\langle m,k\rangle$, 其中 $m\in\mathcal{M}$ 是要被签署的消息, $k\in\mathcal{K}$ 是一个 keystone 的承诺, 然后 $\mathcal{S}$ 按如下步骤回答签名提问:

1) $\mathcal{S}$ 随即选取 c 和 $e\in\mathbb{Z}_q$, 其中 e 不能等于 H 预言机先前的任何输出;

2) $\mathcal{S}$ 计算 $r=g^cy^e \bmod p$. 如果 $\Sigma=\langle m,k,r\rangle$ 是 H 预言机先前的某个输出, 则返回上一步;

3) $\mathcal{S}$ 将二元组 $\langle\Sigma,e\rangle$ 添加到 H-List;

4) $\mathcal{S}$ 输出 $s=\langle r,e,c\rangle$ 作为消息 m 的数字签名.

注意 这里必须检查一下真实的数字签名 δ 和伪造的数字签名 δ' 在概率分布上是否相同.

$$\begin{cases}\delta=\{(r,e,c)|k\in\mathbb{Z}_q,k\neq 0,e\in\mathbb{Z}_q,r=g^k \bmod p,c=k+xe \bmod q\},\\ \delta'=\{(r,e,c)|e\in\mathbb{Z}_q,c\in\mathbb{Z}_q,r=g^cy^e\neq 1 \bmod p\}.\end{cases}\tag{3}$$

首先, 计算一下通过密钥签署的真实数字签名的概率分布:

$$\Pr_{\delta}[(r,e,c)=(\varepsilon,\beta,\gamma)]=\Pr_{k\neq 0,e}[r=g^k=\varepsilon,e=\beta,c=k+xe=\gamma]=\frac{1}{q(q-1)}\tag{4}$$

下面是伪造的数字签名的概率分布:

$$\Pr_{\delta'}[(r,e,c)=(\varepsilon,\beta,\gamma)]=\Pr_{e,c}[e=\beta,c=\gamma,r=g^cy^e=\varepsilon\neq 1 \bmod p]=\frac{1}{q(q-1)}.\tag{5}$$

所以即使不知道密钥也可以以一个不可区分的概率分布模拟仿真出三元组 $\langle r,e,c\rangle$. 因此, 由 $\mathcal{S}$ 所模拟出的签名预言机是高质量的, 进而 $\mathcal{A}$ 对所模拟出的 Sign-Queries 的回答也会非常满意. 他就能够充分施展他的伪造能力.

输出 最后, $\mathcal{A}$ 以一个不可忽略的概率对一个输入消息 $m\in\mathcal{M}$ 和 $k\in\mathcal{K}$ 输出一个签名消息 $s=\langle r,e,c\rangle$, 而且满足 $\mathrm{SVerify}_y(m,k,s)=\mathrm{True}$. 这里, $\mathcal{A}$ 是通过对 KGen 的提问生成了 $k=G(\mathrm{keystone})$. 而且 $\mathcal{A}$ 向 KReveal 询问过 k 但是并没有向 Sign 询问 $\langle m,k\rangle$.

现在 $\mathcal{S}$ 将上述过程模拟仿真两次, 所以 $\mathcal{A}$ 将能够得到两个合法的签名 $s=\langle r,e,c\rangle$ 和 $s'=\langle r,e',c'\rangle$, 其中 $e\neq e'$. 那么就能够得到以下的等式:

$$r=g^cy^e=g^{c-xe}=g^{c'-xe'}=g^{c'}y^{e'} \bmod p.\tag{6}$$

由 (6) 式 $\mathcal{S}$ 就能够在平均期望时间

$$\frac{120686\times 2^q\times Q_HT}{10\times(R+1)\times(R+Q_H)}$$

内解决困难的离散对数问题:

$$\log_g y=-x=\frac{c-c'}{e'-e} \bmod q,\tag{7}$$

这与离散对数问题的困难性相矛盾.

在文献 [22] 中, 为了让两个参与者的签名消息同时发生效力, Chen 等充分利用了环签名具有模糊性的特点. 除了签名的初始发起者, 任何人都不能够确定在两个签名者中究竟谁是某个签名的签名者, 直到签名的初始发起者公布 keystone. 但是在 FESS 方案中, 为了签名生效的同时性, 我们引入了沉睡性这一性质: 两个已经签署并交换的沉睡签名不会发生法律效力, 直到安全承诺信息 keystone 公开并激活签名以后才会发生法律效力.

引理 3 (沉睡性) FESS 在公布安全承诺信息 keystone 之前是满足沉睡性.

证明 首先随机预言机的假设与引理 2 相同. 假设存在一个概率多项式时间图灵机 $\mathcal{A}$, 它的输入包括所有的公开数据. 假设 $\mathcal{A}$ 它可以向随机预言机 G 询问 Q_G 个问题, 向随机预言机 H 询问 Q_H 个问题, 向随机预言机 KGen 询问 Q_K 个问题, 向签名预言机 Sign 询问 R 个问题.

模拟仿真 $\mathcal{S}$ 把公共参数 $\langle g,p,q\rangle$ 和 $y=g^{-x} \bmod p$ 给 $\mathcal{A}$. $\mathcal{S}$ 试图通过模拟所有的预言机来找到由 $k=G(\text{keystone})$ 所对应的 keystone 来模拟挑战者. $\mathcal{A}$ 可以像引理 2 中定义的那样来提问问题.

输出 最后, $\mathcal{A}$ 以一个不可忽略的概率对一个输入消息 $m\in\mathcal{M}$ 和 $k\in\mathcal{K}$ 输出一个 keystone 和一个签名消息 $s=\langle r,e,c\rangle$, 而且满足 $\text{KVerify}_y(m,k,s,\text{keystone})=\text{True}$. 在这种情况下 $\mathcal{A}$ 通过对 KGen 的问询生成一个 $k=G(\text{keystone})$, 而且 $\mathcal{A}$ 没有向 KReveal 询问过 k.

在这个模型下, 要想通过 Sign-Queries 获得一个合法的签名 $s=\langle r,e,c\rangle$ 并且满足 $\text{SVerify}_y(m,k,s)=\text{True}$, 对于 $\mathcal{A}$ 来说是一件很容易的工作. 但是 $\mathcal{A}$ 并不能向 KReveal-Queries 提问问题, 所以他只能够以一个可以忽略的概率 Q_GQ_K/q^2 来猜出 keystone. 这一点与我们最初的假设: $\mathcal{A}$ 以一个不可忽略的概率对一个输入消息 $m\in\mathcal{M}$ 和 $k\in\mathcal{K}$ 输出一个 keystone 和一个签名消息 $s=\langle r,e,c\rangle$, 而且满足 $\text{KVerify}_y(m,k,s,\text{keystone})=\text{True}$ 相矛盾.

引理 4 (公平性) FESS 满足公平性.

证明 首先随机预言机的假设与引理 2 相同. 假设存在一个概率多项式时间图灵机 $\mathcal{A}$, 它的输入包括所有的公开数据. 假设 $\mathcal{A}$ 它可以向随机预言机 G 询问 Q_G 个问题, 向随机预言机 H 询问 Q_H 个问题, 向随机预言机 KGen 询问 Q_K 个问题, 向签名预言机 Sign 询问 R 个问题.

模拟仿真 $\mathcal{S}$ 把公共参数 $\langle g,p,q\rangle$ 和 $y=g^{-x} \bmod p$ 给 $\mathcal{A}$. $\mathcal{S}$ 试图通过模拟所有的预言机来获得密钥 x 或者找到 $k=G(\text{keystone})$ 所对应的 keystone 来模拟挑战者. $\mathcal{A}$ 可以像我们先前定义的那样来提问问题.

输出 最后, $\mathcal{A}$ 以一个不可忽略的概率对一个输入消息 $m\in\mathcal{M}$ 和 $k\in\mathcal{K}$ 输出一个 keystone 和一个签名消息 $s=\langle r,e,c\rangle$, 而且满足 $\text{KVerify}_y(m,k,s,\text{keystone})=\text{True}$. 这时下面的两种情况必然有一个发生:

1) $\mathcal{A}$ 通过对 KGen 的提问生成 $k=G(\text{keystone})$. 而且 $\mathcal{A}$ 向 KReveal 询问过 k 但是并没有向 Sign 询问过 $\langle m,k\rangle$.

2) $\mathcal{A}$ 通过对 KGen 的提问生成 $k=G(\text{keystone})$. 但是 $\mathcal{A}$ 并没有向 KReveal 询问过 k. 在第 1 种情况下, 很容易由引理 2 推导出矛盾. 在第 2 种情况下, 输出的条件只是在一个可以忽略的概率下发生, 这一种情况可以由引理 3 推导得出.

定理 5 FESS 在随机预言机模型下是安全的, 假设求解离散对数问题是困难的.

证明 对于该定理的证明可以直接由正确性、不可伪造性、沉睡性和公平性直接得出.

5.2 执行效率

因为 FESS 能够由更为简单高效的签名方案来实现, 它有着比同时生效签名更为高的执行效率. 表 1 给出了 FESS 和同时生效签名的执行效率比较. 在表 1 中, "E" 表示在 $\mathbb{Z}_p$ 上的指数运算次数, "M_p" 表示在 $\mathbb{Z}_p$ 上的乘法运算次数, "M_q" 表示在 $\mathbb{Z}_q$ 上的乘法运算次数, "A" 表示在 $\mathbb{Z}_q$ 上的加法运算次数, "H" 表示 Hash 算法的执行次数.

表 1　执行效率比较

算法	FESS	同时生效签名
Initial Sign	$1E+1M_q+1A+2H$	$2E+1M_q+1M_p+2A+2H$
Respond Sign	$1E+1M_q+1A+1H$	$2E+1M_q+1M_p+2A+1H$
SVerify	$2E+1M_p+1H$	$3E+2M_p+1A+1H$
KVerify	$2E+1M_p+2H$	$3E+2M_p+1A+2H$

6　结论

本文提出了一种安全并且高效的公平交换数字签名方案 — FESS. 在该方案中, 两个参与者以一种公平的方式交换数字签名. 由于 FESS 是一种构造公平交换签名方案的基本模型, 所以它可以由现在已有的多数签名方案来实现. 在 FESS 方案中, 两个未被激活的签名可以很容易被其他人验证, 但是它并不具有合法的效力, 直到一条额外的信息 — keystone 被其中的一个参与者公开. 一旦 keystone 被公开, 两个签名同时被激活从而变得有效. 我们所提出的方案的一个显著的特征是两个参与者通过一个安全的承诺信息来同时交换数字签名, 而不需要任何可信第三方的介入. 虽然 FESS 没有能够克服同时生效签名由发起方控制 keystone 的弱点, 但是从两种签名方案的性能比较中, 我们可以得出 FESS 的执行效率要高于同时生效签名. 因为 FESS 可以从已有的多数数字签名机制得到许多不同的实现方案, 所以它可以变得能够应用于各种不同的环境. 作为一种有效的密码工具, FESS 为构造公平电子支付协议和公平电子合同签署协议提供了新的思路.

FESS 方案可以很容易拓展到多方参与的情况. 安全性假设可以用同样的方式进行证明. 我们下一步的研究方向是如何减少签名的发起者对 keystone 的公布优势.

参考文献

1 Brickell E F, Chaum D, Damgard I B, et al. Gradual and verifiable release of a secret. In: Proc Crypto'87. LNCS, Vol 293. Berlin: Springer-Verlag, 1987. 156–166

2 Ben-Or M, Goldreich O, Micali S, et al. A fair protocol for signing contracts. IEEE Trans Inf Theor, 1990, 36: 40–46

3 Boneh D, Naor M. Timed commitments (extended abstract). In: Proc Crypto'00. LNCS, Vol 1880. Berlin: Springer-Verlag, 2000. 236–254

4 Cleve R. Controlled gradual disclosure schemes for random bits and their applications. In: Proc Crypto'89. LNCS, Vol 435. Berlin: Springer-Verlag, 1989. 573–588

5 Damgard I B. Practical and provably secure release of a secret and exchange of signatures. In: Proc Eurocrypt'93. LNCS, Vol 765. Berlin: Springer-Verlag, 1993. 200–217

6 Goldreich O. A simple protocol for signing contracts. In: Proc Crypto'83. New York: Plenum Press, 1984. 133–136

7 Garay J, Pomerance C. Timed fair exchange of standard signatures. In: Proc Financial Cryptography 2003. LNCS, Vol 2742. Berlin: Springer-Verlag, 2003. 190–207

8 Asokan N, Shoup V, Waidner M. Optimistic fair exchange of digital signatures. In: Proc Eurocrypt'98. LNCS, Vol 1403, Berlin: Springer-Verlag, 1998. 591–606

9 Asokan N, Shoup V, Waidner M. Optimistic fair exchange of signatures. IEEE J Sel Area Commun, 2000, 18: 593–610

10 Boyd C, Foo E. Off-line fair payment protocols using convertible signature. In: Proc Asiacrypt'98. LNCS, Vol 1514. Berlin: Springer-Verlag, 1998. 271–285

11 Bao F. Colluding attacks to a payment protocol and two signature exchange schemes. In: Proc Asiacrypt'04. LNCS, Vol 3329. Berlin: Springer-Verlag, 2004. 417–429

12 Bao F, Deng R H, Mao W. Efficient and practical fair exchange protocols with off-line TTP. In: Proceedings of IEEE Symposium on Security and Privacy. Los Alamitos: IEEE Computer Society, 1998. 77–85
13 Boneh D, Gentry C, Lynn B, et al. Aggregrate and verifiably encrypted signatures from bilinear maps. In: Proc Eurocrypt'03. LNCS, Vol 2656. Berlin: Springer-Verlag, 2003. 416–432
14 Deng R H, Gong L, Lazar A A, et al. Practical protocols for certified electronic mail. J Netw Syst Manage, 1996, 4: 279–297
15 Dodis Y, Reyzin L. Breaking and repairing optimistic fair exchange from PODC 2003. In: Proceedings of ACM Workshop on Digital Rights Management (DRM). New York: ACM Press, 2003. 47–54
16 Franklin M, Reiter M. Fair exchange with a semi-trusted third party. In: Proceedings of the 4th ACM Conference on Computer and Communications Security. New York: ACM Press, 1997. 1–6
17 Garay J, Jakobsson M, MacKenzie P. Abuse free optimistic contract signing. In: Proc Crypto'99. LNCS, Vol 1666. Berlin: Springer-Verlag, 1999. 449–466
18 Park J M, Chong E, Siegel H, et al. Constructing Fair-Exchange Protocols for E-Commerce Via Distributed Computation of RSA Signatures. In: Proceedings of the 22nd ACM Symposium on Principles of Distributed Computing (PODC 2003). New York: ACM Press, 2003. 172–181
19 Zhou J, Gollmann D. A fair non-repudiation protocol. In: Proceedings of IEEE Symposium on Security and Privacy. Los Alamitos: IEEE Computer Society, 1996. 55–61
20 Zhou J, Gollmann D. An efficient non-repudiation protocol. In: Proceedings of 10th IEEE Computer Security Foundations Workshop. Los Alamitos: IEEE Computer Society, 1997. 126–132
21 Zhou J, Deng R, Bao F. Some remarks on a fair exchange protocol. In: Proceedings of the 3rd International Workshop on Practice and Theory in Public Key Cryptosystems, PKC 2000. LNCS, Vol 1751. London: Springer-Verlag, 2000. 46–57
22 Chen L, Kudla C, Paterson K G. Concurrent Signature. In: Proc Eurocrypt'04. LNCS, Vol 3027. Berlin: Springer-Verlag, 2004. 287–305
23 Rivest R, Shamir A, Tauman Y. How to leak a secret. In: Proc Asiacrypt'01. LNCS, Vol 2248. Berlin: Springer-Verlag, 2001. 552–565
24 Abe M, Ohkubo M, Suzuki K. 1-out-of-n signatures from a variety of keys. In: Proc Asiacrypt'02. LNCS, Vol 2501. Berlin: Springer-Verlag, 2002. 415–432
25 Susilo W, Mu Y, Zhang F. Perfect concurrent signature schemes. In: Proceedings of Information and Communications Security (ICICS'04). LNCS, Vol 3269. Berlin: Spriger-Verlag, 2004. 14–26
26 Wang G, Bao F, Zhou J. The Fairness of perfect concurrent signatures. In: Proceedings of Information and Communications Security (ICICS'06). LNCS, Vol 4307. Berlin: Spriger-Verlag, 2006. 435–451
27 Pointcheval D, Stern J. Security proofs for signature schemes. In: Proc Eurocrypt'96. LNCS, Vol 1070. Berlin: Springer-Verlag, 1996. 387–398
28 Pointcheval D, Stern J. Security arguments for digital signatures and blind signatures. J Cryptol, 2000, 13: 361–396
29 Bellare M, Rogaway P. Random oracles are practical: a paradigm for designing efficient protocols. In: Proc 1st CCCS. New York: ACM press, 1993. 62–73