

高效 SM2 适配器签名及其应用

摘要: 适配器签名, 又名无脚本脚本, 是解决区块链应用 (如加密货币) 中扩展性差、吞吐量低等问题的重要密码技术. 适配器签名可看作数字签名关于困难关系的扩展, 同时具有签名授权和证据提取两种功能, 在区块链应用中具有以下优点: (i) 降低链上成本, (ii) 提高交易的可替代性, (iii) 突破区块链脚本语言限制. SM2 签名是我国自主设计的国家标准算法, 在各种重要信息系统中有着广泛应用. 在本文中, 我们首次基于国密 SM2 签名构造了适配器签名方案, 并在随机预言机模型下基于 SM2 的安全性给出安全性证明. 随后, 我们根据 SM2 签名的结构特点, 采用离线证明技术进行优化, 获得了可支持离线证明的 SM2 适配器签名方案. 该方案与现有 ECDSA 适配器签名相比更加高效. 最后, 我们基于现有适配器签名应用, 分别给出 SM2 适配器签名在区块链中的两种典型应用: 原子交换协议和支付通道网络.

关键词: SM2 算法; 适配器签名; 区块链; 原子交换; 支付通道网络

中图分类号: TP309.7 **文献标识码:** A **DOI:** 10.13868/j.cnki.jcr.0000XX

An Efficient SM2-based Adaptor Signature and Its Applications

Abstract: Adaptor signature, also known as scriptless scripts, is an important cryptographic technique that can be used to solve the problems of poor scalability and low transaction throughput in blockchain applications such as cryptocurrency. Adaptor signature can be seen as an extension of digital signature on hard relations, and it can tie together the authorization with witness extraction and has many advantages in blockchain applications, such as (i) low on-chain cost, (ii) improved fungibility of transactions, (iii) advanced functionality beyond the limitation of the blockchain's scripting language. SM2 signature is the Chinese national standard signature algorithm and has been widely used in various important information systems. In this paper, we first design the adaptor signature based on SM2 signature algorithm, and prove security based on SM2 security under the random oracle model. Then, based on the structure of SM2 signature, we use offline proof technique to optimize it and get a SM2-based adaptor signature supporting offline proof, which is more efficient than the existing ECDSA-based adaptor signature. Finally, based on existing adaptor signature applications, we show two applications of SM2-based adaptor signature in blockchain: atomic swap protocol and payment channel network.

Key words: SM2; adaptor signature; blockchain; atomic swap; payment channel network

1 引言

自 2009 年比特币 [1] 系统出现, 其底层的区块链结构引起了广泛关注. 区块链巧妙地融合了密码、共识机制、P2P 网络等关键技术, 具有去中心化、不可篡改、匿名性、可追溯性、开放透明等特点, 有着非常广阔的应用前景. 但相关区块链应用 (如加密货币) 也面临着可扩展性差、吞吐量低等关键的应用问

题 [2, 3, 4, 5, 6], 比如: 理论上, 比特币的交易吞吐量约为每秒十几笔, 与信用卡每秒数万笔的交易量相比, 低了三个数量级, 严重限制了区块链技术的广泛应用. 区块链上每笔交易可看作某种脚本构成的应用程序, 脚本语言定义了链上可实现的各种功能, 如比特币等支持签名授权的货币转账功能, 以太坊等提供图灵完备的脚本语言, 能够编码更复杂的交易逻辑实现更加丰富的应用功能. 然而, 愈加丰富的功能需要复杂的脚本支持, 导致链上的存储和矿工验证计算成本增加. 针对上述问题, 支付通道网络 [7] 通过在链上建立通道, 可实现用户之间通过该通道进行任意次数的链下交易, 是目前最有前景和部署最广泛的解决方案之一, 如比特币的闪电网络 [8] 和以太坊的雷电网络 [9] 等. 适配器签名作为构建支付通道网络的关键技术, 是目前解决区块链可扩展性差, 减少链上资源等问题的重要工具.

适配器签名 (adaptor signature, AS), 也称为无脚本脚本 (scriptless scripts), 由 Poelstra [10] 首次提出, 并由 Aumayr 等人 [5] 给出形式化的定义. 适配器签名可看作是数字签名对困难关系 (hard relation) 的扩展. 简而言之, 签名者可以使用签名私钥对消息和困难实例进行预签名, 利用困难关系的证据可将预签名值转化为有效的完整签名值. 同时困难关系的证据可通过预签名值和完整签名值提取. 适配器签名具有以下特性: (i) 只有拥有签名私钥的签名者才能生成预签名; (ii) 只有拥有困难关系证据的用户才能将预签名值转化为完整签名值; (iii) 获得预签名值和完整签名值, 可以验证两者的有效性, 并可提取困难关系的证据.

在具体应用, 比如原子交换协议 [15, 16, 17, 18] 中, 双方可基于适配器签名实现不同币种的公平交换. 一方 (困难关系选择方) 通过选择困难关系 (Y, y) 对交换交易进行预签名, 然后将预签名值 $\hat{\sigma}_1$ 发送给另一方; 另一方根据困难实例 Y 也对交换交易进行预签名, 并将预签名值 $\hat{\sigma}_2$ 返回; 困难关系选择方拥有困难关系的证据 y 可将预签名值 $\hat{\sigma}_2$ 转换成交换交易的全签名值 σ_2 , 在链上公布签名值 σ_2 获得交换的货币; 另一方根据签名值 σ_2 和预签名值 $\hat{\sigma}_1$ 可提取证据 y , 并将预签名值 $\hat{\sigma}_1$ 转换成交换交易的全签名值 σ_1 , 在链上公布签名值 σ_1 获得交换的货币. 至此, 双方完成公平交换交易. 根据上述应用的介绍, 适配器签名通过将签名授权和证据提取巧妙地结合, 完成货币交换的复杂操作, 具有减少链上操作, 提高交易的可替代性和突破区块链脚本语言限制等多种优势, 在许多区块链应用中都显示出了重要的应用价值, 如支付通道网络 [11, 7, 5]、支付通道网络中的支付路由 [12, 13, 14] 和原子交换协议 [15, 16, 17, 18] 等.

SM2 签名算法 [19, 20] 是我国国家密码管理局公布的一种拥有完全自主知识产权的国家标准签名算法. 该算法基于椭圆曲线密码体制, 具有安全性高、签名速度快、占用空间小等优势, 在我国商密领域各类信息系统中有着重要应用. 现有的适配器签名主要基于 Schnorr 签名 [21, 10, 5, 22]、ECDSA 签名 [23, 24, 5, 27], 以及格签名 [25, 26] 等方案构造, 缺少基于 SM2 算法的适配器签名方案, 限制了 SM2 算法在区块链场景下的推广应用. 此外, 现有基于 ECDSA 的适配器签名 [5] 在预签名值生成中需要零知识证明保证预签名的正确执行, 导致其预签名操作复杂, 应用效率较低. 秉承着核心技术自主创新、信息安全自主可控的理念, 本文主要探索基于 SM2 签名设计安全高效的 SM2 适配器签名方案, 为 SM2 算法在区块链场景中提供更加高效契合的应用参考, 为设计安全可控的区块链应用提供保障.

2 本文工作

在本文中, 我们首次基于国密 SM2 签名算法构造适配器签名方案, 并在随机预言机模型 (random oracle model, ROM) 下基于 SM2 签名的安全性给出适配器签名方案的可证明安全. 然后, 我们根据 SM2 签名的独特结构, 采用“离线证明技术”对 SM2 适配器签名进一步优化. 优化后的方案与现有 ECDSA 适配器签名方案 (ECDSA-AS) [5] 相比, 在线预签名阶段的计算效率更高, 预签名尺寸更小. 随后, 我们对 SM2 适配器签名和 ECDSA 适配器签名进行理论和实验对比, 发现优化后的 SM2 适配器签名的在线预签名操作需要更少的点乘运算, 耗时仅为 ECDSA 适配器签名的 $1/4$. 我们对本文工作介绍如下:

SM2 适配器签名. SM2 适配器签名是 SM2 签名和困难关系的扩展, 主要包含了预签名算法、预签名验证算法、适配算法、证据提取算法, 以及 SM2 的密钥生成算法、签名算法和签名验证算法. 其中, 预签名算法根据签名私钥和困难实例对消息进行预签名生成预签名值; 预签名验证算法可验证该预签名值的正确性; 适配算法可根据困难关系的证据将该预签名值转化为完整 SM2 签名值; 提取算法可根据预签名值和完整 SM2 签名值提取困难关系的证据. 具体而言, 令 (X, x) 为 SM2 签名算法的公私钥对,

$(I = (Y, \pi), y) \in R_I$ 是离散对数困难关系 [5]. 签名方可计算预签名值: 计算点 $Z = (x+1)Y$, 证明 X 和 $Z - Y$ 具有相同的离散对数关系 ($Z - Y = xY, X = xG$): $\pi_Z \leftarrow P((G, X, Y, Z - Y), x)$, 选择随机数 k 计算 $\hat{K} = kG + Z = (k + y(x+1))G = \hat{k}G = (r_x, r_y)$, $r = r_x + h(m)$, $\hat{s} = (1+x)^{-1}(k - r \cdot x)$, 即预签名值 $\hat{\sigma} = (r, \hat{s}, Z, \pi_Z)$. 适配器算法可基于证据 y 将 $\hat{\sigma}$ 转化为完整 SM2 签名值 $\sigma = (r, s)$, 其中 $s = \hat{s} + y = (x+1)^{-1}(k + y(x+1) - r \cdot x) = (x+1)^{-1}(\hat{k} - r \cdot x)$.

可证明安全. SM2 适配器签名采用具有在线提取器性质的非交互零知识证明 [28] 提供的“自证明结构”[5] 保证可证明安全. 具体而言, $(I = (Y, \pi), y) \in R_I$ 是具有“自证明结构”的离散对数困难关系 [5], 其中 $\pi \leftarrow P(Y, y)$ 证明存在证据 y 使得 $Y = y \cdot G$. 在安全性证明中, 根据困难关系的“自证明结构”, 归约算法在随机预言机模型下能提取证据 y , 然后可基于 SM2 签名预言机输出的完整签名值 $\sigma = (r, s)$, 并通过计算 $\hat{s} = s - y$ 模拟 SM2 适配器签名的预签名预言机. 因此, 如果存在敌手可伪造 SM2 适配器签名的预签名值 $\hat{\sigma}^* = (r^*, \hat{s}^*)$, 则归约算法可根据提取的证据 y 伪造 SM2 的签名值 $\sigma^* = (r^*, s^*)$, $s^* = \hat{s}^* + y$, 打破 SM2 签名的安全性, 从而证明 SM2 适配器签名的安全性基于 SM2 签名的安全性.

离线证明优化技术. SM2 适配器签名和现有 ECDSA 适配器签名 ECDSA-AS [5] 都需要使用零知识证明保证预签名的正确执行. 不同的是, SM2 适配器签名的零知识证明与消息值及预签名使用的随机数无关. 因此, 零知识证明部分可以离线执行, 提升在线预签名操作的效率. 具体而言, 在预签名过程中, ECDSA-AS 需要以预签名的随机数作为证据, 证明离散对数相等的困难关系 $\{((G, K, Y, \hat{K}), k) \mid k \in \mathbb{Z}_n, \text{ s.t. } \hat{K} = kG \wedge K = kY\}$. 因此, 该证明只能由预签名方生成, 而且多次预签名计算需要选用不同的随机数, 对于新的随机数需要进行新的零知识证明.

根据 SM2 签名的结构特点, 预签名过程可使用预签名私钥 x 作为证据, 证明离散对数相等的困难关系 $\{((G, X, Y, Z), x) \mid x \in \mathbb{Z}_n, \text{ s.t. } X = xG \wedge Z - Y = xY\}$. 同时根据 $Z = (x+1)Y = y(X+G)$ 的结构, 也能使用 y 为证据, 证明离散对数相等的困难关系 $\{((G, Y, X+G, Z), y) \mid y \in \mathbb{Z}_n, \text{ s.t. } Z = y(X+G) \wedge Y = yG\}$. 根据 Z 结构的两种证明方式, 可对应 SM2 适配器签名更丰富且灵活的应用: (i) 以预签名私钥 x 为证据时, 关于 Z 结构的证明只能由预签名方生成 (类似 ECDSA-AS); (ii) 以 y 为证据时, 关于 Z 结构的证明可由困难关系选择方生成. 因此, 我们可采用“离线证明技术”对 SM2 适配器签名进行优化, 设计更加高效的 SM2 适配器签名方案. 令 SM2-ASv1 表示 SM2 适配器签名方案, SM2-ASv2 表示优化后的 SM2 适配器签名方案. SM2-ASv2 采用离线证明技术并不影响其“自证明结构”, 可保证在安全性证明中归约算法提取证据 y . 因此, 上述优化提升 SM2 适配器签名预签名阶段的效率, 不影响 SM2-ASv2 的安全性.

性能评估. 我们分别通过理论和实验分析, 对现有 ECDSA-AS [27, 5], SM2-ASv1/2 进行对比. 在线预签名阶段, ECDSA-AS 需要证明离散对数相等关系, 而根据 SM2 签名算法特性, SM2-ASv2 可进行离线证明, 因此适配器签名 SM2-ASv2 比 ECDSA-AS [27, 5] 更加高效. 随后, 基于 OpenSSL, 我们实现了 ECDSA-AS [5], SM2-ASv1/2, 对比了三种方案的在线预签名算法和预签名验证算法的计算效率. 实验结果显示 SM2-ASv2 的在线预签名耗时仅为 ECDSA-AS [5] 的 1/4, 实验结果符合具体理论分析.

应用推广. 在现有适配器签名应用基础上 [16, 13, 26, 5], 我们分别给出 SM2 适配器签名在区块链中的两种典型应用: 原子交换协议 [15, 16, 17, 18] 和支付通道网络 [11, 7, 5], 分别实现跨链货币公平交换功能和支付通道网络下多跳支付功能, 为 SM2 适配器签名的具体应用推广提供参考.

3 预备知识

3.1 符号

本文中, 用 \mathbb{N} 表示自然数集, 用 $\lambda \in \mathbb{N}$ 表示安全参数. 对任意 $n \in \mathbb{N}$, 符号 $[1, n]$ 表示集合 $\{1, \dots, n\}$. 对任意有限集 X , $x \leftarrow X$ 表示从 X 中均匀随机地选取出一个元素 x .

关于安全参数 λ 的函数 negl , 如果 $\text{negl}(\lambda) \geq 0$, 而且对任意正多项式 poly , 存在正整数 $c \in \mathbb{N}$, 使得对于所有的 $n \geq c$ 满足 $\text{negl}(n) \leq \frac{1}{\text{poly}(n)}$, 我们称函数 negl 是可忽略的. 对任意概率算法 \mathcal{A} , 记 \mathcal{A} 所用的

随机数集合为 R . $y \leftarrow \mathcal{A}(x_1, \dots, x_t; r)$ 表示 \mathcal{A} 以 (x_1, \dots, x_t) 和随机数 $r \leftarrow R$ 为输入, 输出 y ; 通常我们忽略 r , 记为 $y \leftarrow \mathcal{A}(x_1, \dots, x_t)$. 如果 \mathcal{A} 的运行时间是关于安全参数 λ 的多项式, 则称 \mathcal{A} 为概率多项式时间 (probabilistic polynomial time, PPT) 算法.

3.2 签名

签名方案 [5] 包含三个多项式时间算法 $\Sigma = (\text{Gen}, \text{Sign}, \text{Vrfy})$, 具体定义如下:

- $\text{Gen}(1^\lambda)$: 密钥生成算法输入安全参数 λ , 输出签名密钥对 (vk, sk) .
- $\text{Sign}(sk, m)$: 签名算法输入签名私钥 sk 和消息 $m \in \{0, 1\}^*$, 输出签名值 σ .
- $\text{Vrfy}(vk, m, \sigma)$: 验证算法输入验证公钥 vk , 消息 $m \in \{0, 1\}^*$ 和签名值 σ , 输出 1 代表签名正确, 否则输出 0.

正确性. 对任意 $\lambda \in \mathbb{N}$, $(vk, sk) \leftarrow \text{Gen}(1^\lambda)$, $m \in \{0, 1\}^*$, 则 $\Pr[\text{Vrfy}(vk, m, \text{Sign}(sk, m)) \rightarrow 1] = 1$.

SUF-CMA 安全. 如果对于任意 PPT 敌手 \mathcal{A} , 存在可忽略函数 negl , 使得在实验 $\text{sSigForge}_{\Sigma, \mathcal{A}}$ 中, 敌手优势 $\text{Adv}_{\mathcal{A}, \text{sSigForge}} = \Pr[\text{sSigForge}_{\Sigma, \mathcal{A}}(\lambda) = 1] \leq \text{negl}(\lambda)$, 则称签名方案 Σ 是选择消息攻击下强不可伪造 (strong existential unforgeability under chosen message attack, SUF-CMA), 其中实验 $\text{sSigForge}_{\Sigma, \mathcal{A}}$ 定义如下:

$\text{sSigForge}_{\mathcal{A}, \Sigma}(\lambda)$	$\mathcal{O}_S(m)$
$\mathcal{Q} = \emptyset$	$\sigma \leftarrow \text{Sign}(sk, m)$
$(vk, sk) \leftarrow \text{Gen}(1^\lambda)$	$\mathcal{Q} = \mathcal{Q} \cup \{m, \sigma\}$
$(m, \sigma) \leftarrow \mathcal{A}^{\mathcal{O}_S(\cdot)}(vk)$	return σ
return $((m, \sigma) \notin \mathcal{Q} \wedge \text{Vrfy}(vk, m, \sigma))$	

3.3 SM2 签名

令 \mathbb{G} 为椭圆曲线上的群, G 为群 \mathbb{G} 中阶为 n 的基点. 随机选择签名私钥 $x \in [1, n-2]$, 计算验证公钥 $X = xG$. 为了方便, 本文省略 SM2 的编码部分, 具体可见 SM2 签名标准 [19].

关于消息 $m \in \{0, 1\}^*$ 的 SM2 签名步骤如下:

1. 计算 $e = H(m)$ ¹.
2. 选择随机数 $k \in [1, n-1]$, 计算 $r_x = f(kG)^2$.
3. 计算 $r = r_x + e \bmod n$, 如果 $r = 0$ 或者 $r + k = n$ 则返回步骤 2.
4. 计算 $s = (1 + x)^{-1}(k - rx) \bmod n$, 如果 $s = 0$ 则返回步骤 2.
5. 输出签名值 $\sigma = (r, s)$.

SM2 签名 $\sigma = (r, s)$ 的验证步骤如下:

1. 如果 $r \notin [1, n-1]$ 或者 $s \notin [1, n-1]$, 输出 0 并退出.
2. 计算 $e = H(m)$.
3. 计算 $t = (r + s) \bmod n$, 如果 $t = 0$, 输出 0 并退出.
4. 计算 $r'_x = f(sG + tX)$.
5. 计算 $r' = r'_x + e \bmod n$, 如果 $r' = r$ 则输出 1 否则输出 0.

SM2 签名算法是我国自主研发的签名算法, 具有安全性高, 运算速度快, 签名尺寸小等优势. SM2 签名算法的安全性分析已有较多工作, 具体可参考 [29, 20, 19].

¹其中 $H(\cdot)$ 表示哈希函数, 具体可见 SM2 签名标准 [19]

²SM2 的变换函数 f 与 ECDSA 类似, 被定义为点 kG 的横坐标, 即 $kG = (r_x, r_y)$, $r_x = f(kG)$ [29, 5].

3.4 困难关系和零知识证明

令实例/证据对 (Y, y) 是一对关系, 定义语言 $L_R = \{Y \mid \exists y, \text{s.t.} (Y, y) \in R\}$. R 是困难关系 [5], 如果以下条件成立:

- $\text{GenR}(1^\lambda) \rightarrow (Y, y)$: 存在 PPT 取样算法 GenR 输入 1^λ , 输出关系 R 上的实例/证据对 $(Y, y) \in R$.
- 该关系是多项式时间可判定的.
- 对任意 PPT 敌手 \mathcal{A} , 输入困难实例 Y 输出证据 y 的概率是可忽略的.

对于困难关系 R , 一组算法 (P, V) 是在随机预言机 \mathcal{H} 下具有在线提取器性质的非交互零知识知识证明 (non-interactive zero-knowledge proof of knowledge, NIZK-PoK) [5, 28], 需要满足以下条件:

- 完备性: 对于任意困难关系 $(Y, y) \in R$, $\pi \leftarrow P^{\mathcal{H}}(Y, y)$, 存在可忽略函数 negl , 使得 $\Pr[V^{\mathcal{H}}(Y, \pi) = 1] \geq 1 - \text{negl}(\lambda)$.
- 零知识性: 对于任意困难关系 $(Y, y) \in R$, 存在 PPT 的模拟器 \mathcal{S} 输入实例 Y 可模拟证明 π .
- 在线提取器: 存在 PPT 提取算法 K , 输入实例和证明 (Y, π) 可访问随机预言机的询问序列以及对应回复, 能提取证据 y , 满足 $(Y, y) \in R$.

3.5 适配器签名

关于困难关系 $R = (Y, y)$ 和签名 $\Sigma = (\text{Gen}, \text{Sign}, \text{Vrfy})$ 的适配器签名 $\Sigma_{AS} = (\text{pSign}, \text{pVrfy}, \text{Adapt}, \text{Ext})$ [5], 定义如下:

- $\text{pSign}(pk, sk, m, Y) \rightarrow \hat{\sigma}$. 预签名算法输入公钥 pk 、签名私钥 sk 、消息 $m \in \{0, 1\}^*$, 以及实例 Y , 输出预签名值 $\hat{\sigma}$.
- $\text{pVrfy}(pk, m, Y, \hat{\sigma}) \rightarrow 0/1$. 预签名验证算法输入公钥 pk 、消息 $m \in \{0, 1\}^*$ 、实例 Y , 以及预签名值 $\hat{\sigma}$, 如果预签名值验证成功输出 1, 否则输出 0.
- $\text{Adapt}(\hat{\sigma}, y) \rightarrow \sigma$. 适配算法输入预签名值 $\hat{\sigma}$ 和证据 y , 输出签名值 σ .
- $\text{Ext}(\sigma, \hat{\sigma}, Y) \rightarrow y$. 提取算法输入签名值 σ , 预签名值 $\hat{\sigma}$ 和实例 Y , 输出证据 y .

适配器签名需要满足原签名的正确性, 还需要满足预签名的正确性, 以及预签名的可适配性. 简单来说, 预签名的正确性可保证签名方诚实地生成关于实例 $Y \in L_R$ 的预签名值. 该预签名值能通过验证, 并且能被适配成一个有效的签名值. 基于该预签名值和适配后的完整签名值能提取出实例 Y 的证据. 该性质保证实例 $Y \in L_R$ 的预签名能被适配成一个有效的签名值, 当且仅当签名方知道实例 Y 的证据 y . 预签名的可适配性保证任意有效的预签名值和实例 Y 的证据 y 可以适配成一个有效的签名值, 即使该实例 Y 、预签名值由预签名方恶意地生成. 因此, 预签名的可适配性比预签名的正确性更强.

预签名正确性. 如果对任意的 $\lambda \in \mathbb{N}$, $m \in \{0, 1\}^*$, $(Y, y) \in R$ 有以下式子成立, 则称适配器签名满足预签名正确性.

$$\Pr \left[\begin{array}{c} \text{pVrfy}(pk, \hat{\sigma}, Y, m) = 1 \\ \wedge \\ \text{Vrfy}(pk, \sigma, m) = 1 \\ \wedge \\ (Y, y') \in R \end{array} \middle| \begin{array}{c} \text{Gen}(1^\lambda) \rightarrow (sk, pk) \\ \text{pSign}((pk, sk), Y, m) \rightarrow \hat{\sigma} \\ \text{Adapt}(\hat{\sigma}, y) \rightarrow \sigma \\ \text{Ext}(\sigma, \hat{\sigma}, Y) \rightarrow y' \end{array} \right] = 1$$

预签名可适配性. 如果对任意的 $\lambda \in \mathbb{N}$, $m \in \{0, 1\}^*$, $(Y, y) \in R$, $(pk, sk) \leftarrow \text{Gen}(1^\lambda)$, 以及任意可验证成功的预签名 $\hat{\sigma}$, $\text{pVrfy}(pk, m, Y, \hat{\sigma}) \rightarrow 1$, 都有 $\Pr[\text{Vrfy}(pk, m, \text{Adapt}(\hat{\sigma}, y)) \rightarrow 1] = 1$, 则称适配器签名满足预签名可适配性.

适配器签名需要满足原签名方案的选择消息攻击下的存在不可伪造性 (existential unforgeability under chosen message attack, EUF-CMA), 还需要满足选择消息攻击下适配器签名的存在不可伪造性 (EUF-CMA for adaptor signature, aEUF-CMA), 以及适配器签名的证据可提取性 (witness extractability for adaptor signature, aWE). 具体安全性定义描述如下:

选择消息攻击下适配器签名的存在不可伪造性. 对于任意的 PPT 敌手 \mathcal{A} , 如果存在可忽略函数 negl , 使得敌手优势 $\text{Adv}_{\mathcal{A}, \text{aSigForge}} = \Pr[\text{aSigForge}_{\mathcal{A}, \Pi_{R, \Sigma}}(\lambda) = 1] \leq \text{negl}(\lambda)$ 成立, 则称适配器签名满足 aEUF-CMA, 其中 $\text{aSigForge}_{\mathcal{A}, \Pi_{R, \Sigma}}$ 定义如下:

$\text{aSigForge}_{\mathcal{A}, \Pi_{R, \Sigma}}(\lambda)$	$\mathcal{O}_S(m)$
$Q = \emptyset$	$\sigma \leftarrow \text{Sign}(sk, m)$
$(pk, sk) \leftarrow \text{Gen}(1^\lambda)$	$Q = Q \cup \{m\}$
$m \leftarrow \mathcal{A}^{\mathcal{O}_S(\cdot), \mathcal{O}_{PS}(\cdot)}(pk)$	return σ
$(Y, y) \leftarrow \text{GenR}(1^\lambda)$	
$\hat{\sigma} \leftarrow \text{pSign}((pk, sk), Y, m)$	$\mathcal{O}_{PS}(m, Y)$
$\sigma \leftarrow \mathcal{A}^{\mathcal{O}_S(\cdot), \mathcal{O}_{PS}(\cdot)}(\hat{\sigma}, Y)$	$\hat{\sigma} \leftarrow \text{pSign}((pk, sk), m, Y)$
return $(m \notin Q \wedge \text{Vrfy}(pk, m, \sigma))$	$Q = Q \cup \{m\}$
	return $\hat{\sigma}$

除了提供额外的预签名预言机 $\mathcal{O}_{PS}(\cdot)$, 适配器签名的 aEUF-CMA 安全类似 EUF-CMA. 该预言机输入消息值和实例 Y 输出一个预签名值. 预签名预言机是非常关键的, 因为在应用中适配器签名要求敌手在不知道实例 Y 的证据的情况下, 即使知道预签名值也难以伪造消息的真实签名值. aEUF-CMA 和预签名的可适配性一起保证了关于实例 Y 的预签名值能被转换成一个有效的签名值, 当且仅当知道实例 Y 的证据 y .

证据可提取性. 对于任意的 PPT 敌手 \mathcal{A} , 如果存在可忽略函数 negl , 使得在实验 $\text{aWitExt}_{\mathcal{A}, \Pi_{R, \Sigma}}$ 中敌手优势 $\text{Adv}_{\mathcal{A}, \text{aWitExt}} = \Pr[\text{aWitExt}_{\mathcal{A}, \Pi_{R, \Sigma}}(\lambda) = 1] \leq \text{negl}(\lambda)$, 则称适配器签名满足证据可提取性. $\text{aWitExt}_{\mathcal{A}, \Pi_{R, \Sigma}}$ 定义如下:

$\text{aWitExt}_{\mathcal{A}, \Pi_{R, \Sigma}}(\lambda)$	$\mathcal{O}_S(m)$
$Q = \emptyset$	$\sigma \leftarrow \text{Sign}(sk, m)$
$(pk, sk) \leftarrow \text{Gen}(1^\lambda)$	$Q = Q \cup \{m\}$
$(m, Y) \leftarrow \mathcal{A}^{\mathcal{O}_S(\cdot), \mathcal{O}_{PS}(\cdot)}(pk)$	return σ
$\hat{\sigma} \leftarrow \text{pSign}((pk, sk), Y, m)$	
$\sigma \leftarrow \mathcal{A}^{\mathcal{O}_S(\cdot), \mathcal{O}_{PS}(\cdot)}(\hat{\sigma})$	$\mathcal{O}_{PS}(m, Y)$
$y' \leftarrow \text{Ext}(\sigma, \hat{\sigma}, Y)$	$\hat{\sigma} \leftarrow \text{pSign}((pk, sk), m, Y)$
return $(m \notin Q \wedge (Y, y') \notin R$	$Q = Q \cup \{m\}$
$\wedge \text{Vrfy}(pk, m, \sigma))$	return $\hat{\sigma}$

适配器签名的证据可提取性可保证关于消息和实例 (m, Y) 的一对有效的签名值和预签名值 $(\sigma, \hat{\sigma})$ 能被用于提取出 Y 的证据 y . 实验 aWitExt 和实验 aSigForge 的重要区别: 在实验 aWitExt 中, 敌手可选择实例 Y . 因此, 在实验 aWitExt 中可认为敌手知道实例 Y 的证据 y , 可以生成关于挑战消息 m 的有效签名值. 但是这个优势不足于让敌手赢得实验 aWitExt , 因为敌手赢得实验 aWitExt 的条件是他输出的有效签名值不能泄漏实例 Y 的证据 y .

定义 1 如果适配器签名 Σ_{AS} 满足预签名可适配性, aEUF-CMA 安全和证据可提取性, 则称该适配器签名 Σ_{AS} 是安全的.

4 基于 SM2 的适配器签名

4.1 方案设计

本节中, 我们给出基于 SM2 的适配器签名方案, 并在随机预言机模型下给出安全性证明. 令 $R = \{(I_Y = (Y, \pi_Y), y) \mid Y = yG \wedge \forall (Y, \pi_Y) \rightarrow 1\}$, $R_Z = \{(I_Z = (G, X, Y, Z), x) \mid X = xG \wedge Z - Y = xY\}$, P 和 V 分别表示零知识证明系统中的证明和验证算法. 基于 SM2 的适配器签名算法构造如下:

- $\text{pSign}((X, x), m, I_Y) \rightarrow \hat{\sigma}$. 预签名算法输入公私钥对 (X, x) , 消息值 m 和困难实例 $I_Y = (Y, \pi_Y)$, 计算 $Z = (x+1)Y$, 运行 $\text{P}(I_Z = (G, X, Y, Z), x) \rightarrow \pi_Z$, 计算 $e = H(m)$, 选择随机数 $k \leftarrow \mathbb{Z}_n$, 并计算 $\hat{K} = kG + Z$, $r_x = f(\hat{K})$, $r = e + r_x$, $\hat{s} = (1+x)^{-1}(k - rx) \bmod n$, 最后输出预签名值 $\hat{\sigma} = (r, \hat{s}, Z, \pi_Z)$.
- $\text{pVrfy}(X, m, I_Y, \hat{\sigma}) \rightarrow 0/1$. 预签名验证算法输入公钥 X , 消息 m , 困难实例 $I_Y = (Y, \pi_Y)$, 以及预签名值 $\hat{\sigma} = (r, \hat{s}, Z, \pi_Z)$, 运行 $\text{V}(I_Z, \pi_Z) \rightarrow 0$, 则输出 0, 否则计算 $e = H(m)$, $\hat{K} = (\hat{s}+r)X + \hat{s}G + Z$, $r' = f(\hat{K}) + e \bmod n$. 如果 $r' = r$, 该算法输出 1, 否则输出 0.
- $\text{Adapt}(\hat{\sigma}, y) \rightarrow \sigma$. 适配算法输入预签名值 $\hat{\sigma}$ 和证据 y , 计算 $s = \hat{s} + y \bmod n$, 并输出签名值 $\sigma = (r, s)$.
- $\text{Ext}(\sigma, \hat{\sigma}, I_Y) \rightarrow y$. 证据提取算法输入签名值 σ , 预签名值 $\hat{\sigma}$ 和困难实例 $I_Y = (Y, \pi_Y)$, 计算 $y = s - \hat{s} \bmod n$. 如果 $(Y, y) \in R$, 输出 y , 否则输出 \perp .

$\text{pSign}((X, x), m, I_Y)$	$\text{pVrfy}(X, m, I_Y, \hat{\sigma})$	$\text{Adapt}(\hat{\sigma}, y)$
$pk = X, sk = x$	$pk = X, I_Y = (Y, \pi_Y)$	$\hat{\sigma} = (r, \hat{s}, Z, \pi_Z)$
$I_Y = (Y, \pi_Y)$	$\hat{\sigma} = (r, \hat{s}, Z, \pi_Z)$	$s = \hat{s} + y \bmod n$
$Z = (x+1)Y$	$b_Z \leftarrow \text{V}(I_Z, \pi_Z)$	return $\sigma = (r, s)$
$\text{P}(I_Z, x) \rightarrow \pi_Z$	$e = H(m)$	
$e = H(m)$	$\hat{K} = (\hat{s}+r)X + \hat{s}G + Z$	$\text{Ext}(\sigma, \hat{\sigma}, I_Y)$
$k \leftarrow \mathbb{Z}_n$	$r' = f(\hat{K}) + e \bmod n$	$\sigma = (r, s)$
$\hat{K} = kG + Z, r_x = f(\hat{K})$	$b_r = (r' = r)$	$\hat{\sigma} = (r, \hat{s}, Z, \pi_Z)$
$r = e + r_x$	return $(b_Z \wedge b_r)$	$I_Y = (Y, \pi_Y)$
$\hat{s} = (1+x)^{-1}(k - rx) \bmod n$		$y = s - \hat{s} \bmod n$
return $\hat{\sigma} = (r, \hat{s}, Z, \pi_Z)$		If $(Y, y) \in R$, return y , else return \perp

图 1 基于 SM2 的适配器签名

Figure 1 SM2-based adaptor signature scheme

4.2 安全性证明

定理 1 如果 SM2 签名 Σ_{SM2} 满足 SUF-CMA, 且 R 是一个困难关系, 则上述方案 $\Pi_{R, \Sigma_{\text{SM2}}}$ 在随机预言机模型下是安全的.

引理 1 (预签名的正确性.) 基于 SM2 的适配器签名 $\Pi_{R, \Sigma_{\text{SM2}}}$ 满足预签名的正确性.

证明: 选定 $x, y \in \mathbb{Z}_n$ 和 $m \in \{0, 1\}^*$, 定义 $X = xG, Y = yG, \pi_Y \leftarrow \text{P}((G, Y), y)$ 和 $I_Y = (Y, \pi_Y)$. 根据预签名算法的定义 $\text{pSign}((X, x), m, I_Y) \rightarrow (r, \hat{s}, Z, \pi_Z)$, 对于随机数 $k \leftarrow \mathbb{Z}_n$, $\hat{s} = (1+x)^{-1}(k - rx) \bmod n$. 令 $K = (\hat{s}+r)X + \hat{s}G = kG$.

根据 NIZK_Z 的正确性, 即 $\text{V}(I_Z, \pi_Z) \rightarrow 1$. 因此, $Z = (x+1)Y = (x+1)yG, \hat{K} = K + Z = (k + y(x+1))G, r'_x = f(\hat{K}), r' = r'_x + e = r$, 即 $\text{pVrfy}(X, m, I_Y, \hat{\sigma}) \rightarrow 1$. 由于 $\text{Adapt}(\hat{\sigma}, y) \rightarrow \sigma$, 根据 SM2 签名的正确性 $\text{Vrfy}(pk, m, \sigma) \rightarrow 1$. 因为 $\text{Adapt}(\hat{\sigma}, y) \rightarrow \sigma$, 所以根据适配算法 Adapt 的定义 $s = \hat{s} + y$. 因此, $\text{Ext}((r, s), (r, \hat{s}), I_Y) = \hat{s} + y - \hat{s} = y$. \square

引理 2 (预签名可适配性.) 基于 SM2 的适配器签名 $\Pi_{R, \Sigma_{\text{SM2}}}$ 满足预签名的可适配性.

证明: 对于选定的消息 $m \in \{0, 1\}^*, e = H(m)$, 困难关系 $(I_Y, y) \in R$, 公钥 $X \in \mathbb{G}$ 和有效的预签名值 $\hat{\sigma} = (r, \hat{s}, Z, \pi_Z)$, 因此 $\text{pVrfy}(X, m, I_Y, \hat{\sigma}) \rightarrow 1$. 即 $\hat{K} = (\hat{s}+r)X + \hat{s}G + Z = kG + Z, r' = r'_x + e = f(\hat{K}) + e = f(kG + Z) + e = r$.

根据适配算法的定义, $\text{Adapt}(\hat{\sigma}, y) \rightarrow \sigma$, 其中 $\sigma = (r, s), s = \hat{s} + y = (1+x)^{-1}(k + y(1+x) + r) - r \bmod n$. 因此, $K' = (s+r)X + sG = kG + y(1+x)G = kG + Z$. 即 $r' = r'_x + e = f(K') + e = f(kG + Z) + e = r$. 即预签名值能被适配成 SM2 签名值, 满足 SM2 的签名验证算法 $\text{Vrfy}(pk, m, \sigma) \rightarrow 1$. \square

引理 3 (*aEUF-CMA 安全*.) 如果 SM2 签名算法 Σ_{SM2} 满足 SUF-CMA, 且 R 是一个困难关系, 则基于 SM2 的适配器签名 $\Pi_{R, \Sigma_{\text{SM2}}}$ 满足 aEUF-CMA 安全.

证明: 我们在 ROM 模型下将基于 SM2 的适配器签名的不可伪造性归约到 SM2 签名的强不可伪造性. 假设存在 PPT 敌手 \mathcal{A} , 在 aSigForge 实验中, 打破 SM2 适配器签名的安全性, 则可构造模拟器 \mathcal{S} 打破 SM2 签名的强不可伪造性. \mathcal{S} 可以访问 SM2 签名预言机 $\mathcal{O}_{\text{SM2-Sign}}$ 和随机预言机 \mathcal{H}_{SM2} , 并且模拟 \mathcal{A} 的随机预言机 (\mathcal{H}), 签名预言机 (\mathcal{O}_S) 和预签名预言机 (\mathcal{O}_{PS}).

\mathcal{S} 可以通过 $\mathcal{O}_{\text{SM2-Sign}}$ 和 \mathcal{H}_{SM2} 直接模拟 \mathcal{A} 的 \mathcal{O}_S 和 \mathcal{H} 的询问. 因为 \mathcal{S} 没有预签名私钥无法直接生成预签名值, 所以模拟 \mathcal{A} 的 \mathcal{O}_{PS} 较为困难. 具体挑战包括: 1) \mathcal{S} 需要获得实例 Y 的证据 y . 2) \mathcal{S} 需要模拟在预签名阶段的零知识证明 π_Z . 上述挑战可解决: 当 \mathcal{A} 询问 \mathcal{O}_{PS} 关于消息 m 和实例 $I_Y = (Y, \pi_Y)$ 的预签名值时, 模拟器 \mathcal{S} 先询问自己的 $\mathcal{O}_{\text{SM2-Sign}}$ 获得真实签名, 此外, \mathcal{S} 通过证明 π_Y 提取证据 y , 并根据适配器签名的定义, 基于证据 y 和真实签名值计算预签名值, 最后, 利用零知识性质, 在不知道证据 x 的情况下, 模拟证明 π_Z . 即模拟器 \mathcal{S} 可模拟 \mathcal{A} 的 \mathcal{O}_{PS} .

通过混合论证技术, 我们首先描述 5 个游戏 $\text{Game}_0, \dots, \text{Game}_4$, 其中 Game_0 是真实的 aSigForge 实验, Game_i 和 Game_{i+1} , $i = 0, \dots, 3$ 是不可区分的, 且模拟器可完美的模拟 Game_4 , 最后证明敌手赢得最后游戏 Game_4 的概率是可忽略的.

Game_0 : 该游戏和真实 aSigForge 实验相同.

Game_1 : 该游戏和 Game_0 的区别是当 \mathcal{A} 输出真实签名值伪造 σ^* 时, 如果 $\sigma^* = \text{Adapt}(\hat{\sigma}, y)$, 则输出 \perp .

Game_2 : 该游戏和 Game_1 的区别是在敌手询问 \mathcal{O}_{PS} 时, Game_2 使用 NIZK 的证据提取算法, 基于实例 Y , 证明 π_Y 以及随机预言机 \mathcal{H} 的询问列表提取证据 y , 如果 $((Y, \pi_Y), y) \in R$ 不成立, 则输出 \perp .

Game_3 : 该游戏和 Game_2 的区别是在敌手询问 \mathcal{O}_{PS} 时, Game_3 首先询问 $\mathcal{O}_{\text{SM2-Sign}}$ 获得完整签名值 $\sigma = (r, s)$, 基于证据 y , 计算 $\hat{s} = s - y$, 此外, 计算 $Z = y(X + G)$, 并模拟零知识证明输出证明 $\pi_S \leftarrow S((G, X, Y, Z), 1)$, 输出预签名值 $\hat{\sigma} = (r, \hat{s}, Z, \pi_S)$.

Game_4 : 该游戏和 Game_3 的区别是在接收 \mathcal{A} 输出的挑战消息 m^* 后, Game_4 通过 $\mathcal{O}_{\text{SM2-Sign}}$ 获得完整签名值 $\sigma^* = (r^*, s^*)$, 根据证据 y^* 计算预签名值, 模拟零知识证明输出证明 $\pi_S^* \leftarrow S((G, X, Y^*, Z^*), 1)$, 并输出预签名值 $\hat{\sigma}^* = (r^*, \hat{s}^*, Z^*, \pi_S^*)$.

可构造模拟器 \mathcal{S} 完美模拟 Game_4 , 利用 \mathcal{A} 的能力赢得 sSigForge 实验. 具体构造如下:

- 模拟 \mathcal{A} 的签名预言机: \mathcal{S} 利用自己的签名预言机 $\mathcal{O}_{\text{SM2-sign}}$ 回复 \mathcal{A} 的询问.
- 模拟 \mathcal{A} 的随机预言机: 对于没有询问过的值, \mathcal{S} 利用 \mathcal{H}_{SM2} 回复询问, 对于已经询问过的值, 输出与上次相同的回复.
- 模拟 \mathcal{A} 的预签名预言机: \mathcal{S} 接收到 \mathcal{A} 的 (m, I_Y) , 首先基于 NIZK_Y 的可提取性, 提取证据 y , 并询问自己的预言机 $\mathcal{O}_{\text{SM2-sign}}$ 关于消息 m 的签名 (r, s) . 然后 \mathcal{S} 计算 $\hat{s} = s - y$, $Z = y(G + X)$, 并模拟零知识证明 π_S 证明 $Z - Y = xY$. 最后输出 $\hat{\sigma} = (r, \hat{s}, Z, \pi_S)$.
- 挑战阶段: \mathcal{S} 接收到 \mathcal{A} 的挑战消息 m^* , 运行 $\text{GenR}(1^\lambda) \rightarrow (I_Y, y)$, 询问 $\mathcal{O}_{\text{SM2-sign}}$ 消息 m^* 获得 $\sigma = (r, s)$, 然后计算 $\hat{s} = s - y$, $Z = y(G + X)$, 并模拟零知识证明 π_S 证明 $Z - Y = xY$, 输出 $\hat{\sigma}^* = (r, \hat{s}, Z, \pi_S)$. 最后接收 \mathcal{A} 的伪造 σ^* , 并输出自己的伪造 (m^*, σ^*) .

Claim 1 令事件 Bad_1 指 Game_1 输出 \perp , 则 $\Pr[\text{Bad}_1] \leq \text{negl}_1(\lambda)$.

证明: 我们将事件 Bad_1 归约到困难关系 R 的困难性上, 即假设存在 PPT 敌手 \mathcal{A} 能使得 Game_1 以非可忽略的概率输出 \perp , 则可构造模拟器 \mathcal{S} 打破 R 的困难性. 具体构造如下:

模拟器 \mathcal{S} 获得挑战关系 I_Y^* , 首先生成密钥对 $(pk, sk) \leftarrow \text{Gen}(1^\lambda)$, 并使用 Game_1 中的方式模拟 \mathcal{A} 的 \mathcal{H} , \mathcal{O}_S 和 \mathcal{O}_{PS} . 当收到 \mathcal{A} 的挑战消息 m 时, 计算并输出预签名值 $\hat{\sigma} \leftarrow \text{pSign}((pk, sk), m, I_Y^*)$, 敌手输出完整签名值 σ .

假设事件 Bad_1 发生, 即 $\text{Adapt}(\hat{\sigma}, y) = \sigma$, 因此 \mathcal{S} 可以提取证据 $y^* \leftarrow \text{Ext}(\sigma, \sigma^*, I_Y^*)$, 可获得 y^* 使得 $(I_Y^*, y^*) \in R$ 成立. \mathcal{A} 的视图和在 Game_1 中的视图是不可区分的, 因此, \mathcal{S} 能打破困难关系的概率和 Bad_1 发生的概率相同, 即 $\Pr[\text{Bad}_1] \leq \text{negl}_1(\lambda)$. 因为除了 Bad_1 发生, Game_1 和 Game_0 是相同的, 所以 $|\Pr[\text{Game}_0 = 1] - \Pr[\text{Game}_1 = 1]| \leq \text{negl}_1(\lambda)$. \square

Claim 2 Game_2 , Game_3 和 Game_4 是不可区分的.

证明: 根据非交互零知识证明中在线提取器的正确性, 可基于实例 Y 及证明 π_Y 提取证据 y , 使得 $V(Y, \pi_Y) \rightarrow 1$ 成立, 且 $(I_Y, y) \in R$ 成立. 因此, Game_2 和 Game_1 是不可区分的, 即 $|\Pr[\text{Game}_2 = 1] - \Pr[\text{Game}_1 = 1]| \leq \text{negl}_2(\lambda)$.

根据非交互零知识证明的零知识性, 模拟器 \mathcal{S} 能够模拟证明 π_S , 且 $\pi_S \leftarrow S((G, X, Y, Z), 1)$ 和真实产生的证明 $\pi_Z \leftarrow P((G, X, Y, Z), x)$ 不可区分. 因此, Game_3 和 Game_2 是不可区分的, 即 $|\Pr[\text{Game}_3 = 1] - \Pr[\text{Game}_2 = 1]| \leq \text{negl}_3(\lambda)$.

在 Game_4 中, \mathcal{S} 接收到 \mathcal{A} 的挑战消息 m^* , 首先生成 $(I_Y, y) \leftarrow \text{GenR}(1^\lambda)$, 计算完整签名值 $\sigma = (r, s)$, 和预签名值 $\hat{s} = s - y$, 并模拟证明 $\pi_S \leftarrow S((G, X, Y, Z), 1)$, 输出预签名值 $\hat{\sigma} = (r, \hat{s}, Z, \pi_S)$. 由于非交互零知识证明的零知识性, 模拟生成 π_S 和真实生成 π_Z 不可区分. 因此, Game_4 和 Game_3 是不可区分的, 即 $|\Pr[\text{Game}_4 = 1] - \Pr[\text{Game}_3 = 1]| \leq \text{negl}_4(\lambda)$. \square

Claim 3 (m^*, σ^*) 是实验 sSigForge 中的有效伪造.

证明: 在挑战阶段之前, 敌手 \mathcal{A} 没有询问过 \mathcal{O}_S 或者 \mathcal{O}_{pS} 关于挑战消息 m^* 的签名值或预签名值. 因此, 关于 m^* 对于 $\mathcal{O}_{\text{SM2-Sign}}$ 的询问发生在挑战阶段. 根据 Game_1 可知, 敌手输出的伪造 σ^* 和 $\mathcal{O}_{\text{SM2-Sign}}$ 输出的 σ 相同的概率是可忽略的. 因此, 关于 m^* 的询问, $\mathcal{O}_{\text{SM2-Sign}}$ 从没有输出过 σ^* , 即 (m^*, σ^*) 是实验 sSigForge 中的有效伪造. \square

$\mathcal{S}^{\mathcal{O}_{\text{SM2-Sign}}, \mathcal{H}_{\text{SM2}}}(pk)$	$\mathcal{H}(x)$
$\mathcal{Q} = \emptyset$	if $H[x] = \perp$
$H = [\perp]$	$H[x] \leftarrow \mathcal{H}_{\text{SM2}}(x)$
$m^* \leftarrow \mathcal{A}^{\mathcal{O}_S(\cdot), \mathcal{O}_{\text{pS}}(\cdot)}(pk)$	return $H[x]$
$(I_Y, y) \leftarrow \text{GenR}(1^\lambda)$	
$\sigma \leftarrow \mathcal{O}_{\text{SM2-Sign}}(m^*)$	$\mathcal{O}_{\text{pS}}(m, I_Y)$
parse σ as (r, s)	parse I_Y as (Y, π_Y)
$\hat{s} = s - y$	$y \leftarrow K(Y, \pi_Y, H)$
$Z = y(X + G)$	if $((Y, \pi_Y), y) \notin R$
$\pi_S \leftarrow S((G, X, Y, Z), 1)$	abort
$\hat{\sigma} = (r, s, Z, \pi_S)$	$\sigma \leftarrow \mathcal{O}_{\text{SM2-Sign}}(m)$
$\sigma^* \leftarrow \mathcal{A}^{\mathcal{O}_S(\cdot), \mathcal{O}_{\text{pS}}(\cdot)}(\hat{\sigma}, I_Y)$	parse σ as (r, s)
return (m^*, σ^*)	$\hat{s} = s - y$
	$Z = y(X + G)$
$\mathcal{O}_S(m)$	$\pi_S \leftarrow S((G, X, Y, Z), 1)$
$\sigma \leftarrow \mathcal{O}_{\text{SM2-Sign}}(m)$	$\mathcal{Q} = \mathcal{Q} \cup \{m\}$
$\mathcal{Q} = \mathcal{Q} \cup \{m\}$	return (r, \hat{s}, Z, π_S)
return σ	

综上所述, 可知 $|\Pr[\text{Game}_0 = 1] - \Pr[\text{Game}_4 = 1]| \leq \text{negl}_1(\lambda) + \text{negl}_2(\lambda) + \text{negl}_3(\lambda) + \text{negl}_4(\lambda) \leq \text{negl}(\lambda)$. 即 $\Pr[\text{Game}_0 = 1] \leq \Pr[\text{Game}_4 = 1] + \text{negl}(\lambda)$. 因为 \mathcal{S} 可完美模拟 Game_4 , 所以 \mathcal{A} 在实验 sSigForge 中的优势为:

$$\text{Adv}_{\mathcal{A}, \text{sSigForge}} = \Pr[\text{Game}_0 = 1] \leq \Pr[\text{Game}_4 = 1] + \text{negl}(\lambda) \leq \text{Adv}_{\mathcal{S}, \text{sSigForge}} + \text{negl}(\lambda)$$

因此, 上述 SM2 适配器签名满足 aEUF-CMA 安全. \square

引理 4 (证据可提取性.) 假设 SM2 签名满足 SUF-CMA 安全, 且 R 是一个困难关系, 则适配器签名 $\Pi_{R, \Sigma_{SM2}}$ 满足证据可提取性.

证明: 假设存在 PPT 敌手 \mathcal{A} 能以非可忽略的概率赢得 aWitExt 实验, 则可构造 PPT 的模拟器赢得 SM2 的 sSigForge 实验.

不同于实验 aSigForge, 在实验 aWitExt 中, 由 \mathcal{A} 生成困难关系的实例 I_Y . 因此, \mathcal{S} 无法在挑战阶段生成证据 y , 并通过 SM2 的签名值计算预签名值. 但是 \mathcal{S} 可通过 NIZK 的在线提取器的性质, 提取 I_Y 中的证据 y . \mathcal{S} 同样可模拟敌手的预签名值预言机 \mathcal{O}_{PS} .

同样, 我们定义一系列的游戏 $\text{Game}_0, \dots, \text{Game}_4$, 其中 Game_0 是原 aWitExt 实验, 对于 $i = 0, \dots, 3$, Game_i 和 Game_{i+1} 是不可区分的, 且模拟器可完美的模拟 Game_4 , 最后证明敌手赢得最后游戏 Game_4 的概率是可忽略的.

Game_0 : 该游戏和真实的 aWitExt 实验相同.

Game_1 : 该游戏和 Game_0 的区别是在模拟 \mathcal{O}_{PS} 中通过执行提取算法 K 根据证明 π_Y 提取实例 Y 中的证据 y , 如果不满足 $(I_Y, y) \in R$ 则输出 \perp .

Game_2 : 该游戏和 Game_1 的区别是在模拟 \mathcal{O}_{PS} 中首先生成全签名值 $\sigma = (r, s)$, 然后使用证据 y 去计算 $\hat{s} = s - y$, 然后计算 $Z = y(X + G)$ 并模拟零知识证明计算 π_S , 输出预签名值 $\hat{\sigma} = (r, s, Z, \pi_S)$.

Game_3 : 该游戏和 Game_2 的区别是在挑战阶段使用 Game_1 的方式, 使用 NIZK 的在线提取器 K 提取证据 y , 如果不满足 $(I_Y, y) \in R$ 输出 \perp .

Game_4 : 该游戏和 Game_3 的区别是在挑战阶段接收到挑战 (m, I_Y) 使用 Game_2 中的方式模拟 \mathcal{O}_{PS} , 最后输出预签名值 $\hat{\sigma} = (r, s, Z, \pi_S)$.

可构造模拟器 \mathcal{S} 完美模拟 Game_4 , 利用 \mathcal{A} 的能力赢得 sSigForge 实验. 具体构造如下:

- 模拟 \mathcal{A} 的签名预言机: \mathcal{S} 利用自己的签名预言机 $\mathcal{O}_{SM2\text{-sign}}$ 回复 \mathcal{A} 的询问.
- 模拟 \mathcal{A} 的随机预言机: 对于没有询问过的值, \mathcal{S} 利用 \mathcal{H}_{SM2} 回复询问, 对于已经询问过的值, 输出与上次相同的回复.
- 模拟 \mathcal{A} 的预签名预言机: \mathcal{S} 接收到 \mathcal{A} 的 (m, I_Y) , 首先基于 NIZK $_Y$ 的可提取性, 提取证据 y , 并询问自己的预言机 $\mathcal{O}_{SM2\text{-sign}}$ 关于消息 m 的签名 (r, s) . 然后 \mathcal{S} 计算 $\hat{s} = s - y$, $Z = y(G + X)$, 并模拟零知识证明 π_S 证明 $Z - Y = xY$. 最后输出 $\hat{\sigma} = (r, \hat{s}, Z, \pi_S)$.
- 挑战阶段: \mathcal{S} 接收到 \mathcal{A} 的挑战 (m^*, I_Y) , 首先基于 NIZK $_Y$ 的可提取性, 提取证据 y , 并询问自己的预言机 $\mathcal{O}_{SM2\text{-sign}}$ 关于消息 m^* 的签名 (r, s) . 然后 \mathcal{S} 计算 $\hat{s} = s - y$, $Z = y(G + X)$, 并模拟零知识证明 π_S 证明 $Z - Y = xY$, 输出 $\hat{\sigma} = (r, \hat{s}, Z, \pi_S)$. 最后接收 \mathcal{A} 的伪造 σ^* , 并输出自己的伪造 (m^*, σ^*) .

Claim 4 各个游戏 $\text{Game}_0, \text{Game}_1, \text{Game}_2, \text{Game}_3, \text{Game}_4$ 是不可区分的.

证明: 根据零知识证明的在线提取器的正确性, 模拟器可根据实例 Y 和证明 π_Y 在随机预言机模型下提取证据 y , 使得 $\forall (Y, \pi_Y) = 1$, 且 $((Y, \pi_Y), y) \in R$, 因此, Game_1 和 Game_0 是不可区分的, 即 $|\Pr[\text{Game}_0 = 1] - \Pr[\text{Game}_1 = 1]| \leq \text{negl}_1(\lambda)$.

同样, 在挑战阶段上述性质依旧满足, 因此, Game_2 和 Game_3 是不可区分的, 即 $|\Pr[\text{Game}_2 = 1] - \Pr[\text{Game}_3 = 1]| \leq \text{negl}_3(\lambda)$.

根据非交互零知识证明的零知识性, 在随机预言机模型下, 模拟器 \mathcal{S} 能够模拟证明 π_S , 且 $\pi_S \leftarrow S((G, X, Y, Z), 1)$ 和真实产生的证明 $\pi_Z \leftarrow P((G, X, Z, Y), x)$ 不可区分. 因此, Game_2 和 Game_1 是不可区分的, 即, $|\Pr[\text{Game}_1 = 1] - \Pr[\text{Game}_2 = 1]| \leq \text{negl}_2(\lambda)$.

同样, 在挑战阶段上述性质依旧满足, 因此, Game_3 和 Game_4 是不可区分的, 即 $|\Pr[\text{Game}_3 = 1] - \Pr[\text{Game}_4 = 1]| \leq \text{negl}_4(\lambda)$. \square

Claim 5 (m^*, σ^*) 在 sSigForge 实验中是一个有效的伪造.

证明: 在挑战阶段之前, 敌手 \mathcal{A} 没有询问过 \mathcal{O}_S 或者 \mathcal{O}_{PS} 关于挑战消息 m^* 的签名值或预签名值. 关于 m^* 对于 $\mathcal{O}_{SM2-Sign}$ 的询问发生在挑战阶段. 如果在挑战阶段敌手输出的伪造 σ^* 和 $\mathcal{O}_{SM2-Sign}$ 输出的 σ 相同, 则可提取证据 $y = \hat{s} - s$ 使得 $(I_Y, y) \in R$ 成立. 因此, 关于 m^* 的询问, $\mathcal{O}_{SM2-Sign}$ 从没有输出过 σ^* , 即 (m^*, σ^*) 是实验 sSigForge 中的有效伪造. \square

$\mathcal{S}^{\mathcal{O}_{SM2-Sign}, \mathcal{H}_{SM2}}(pk)$	$\mathcal{H}(x)$
$\mathcal{Q} = \emptyset$	if $H[x] = \perp$
$H = [\perp]$	$H[x] \leftarrow \mathcal{H}_{SM2}(x)$
$(m^*, I_Y) \leftarrow \mathcal{A}^{\mathcal{O}_S(\cdot), \mathcal{O}_{PS}(\cdot)}(pk)$	return $H[x]$
parse I_Y as (Y, π_Y)	
$y \leftarrow K(Y, \pi_Y, H)$	$\mathcal{O}_{PS}(m, I_Y)$
if $((Y, \pi_Y), y) \notin R$	parse I_Y as (Y, π_Y)
abort	$y \leftarrow K(Y, \pi_Y, H)$
$\sigma \leftarrow \mathcal{O}_{SM2-Sign}(m^*)$	if $((Y, \pi_Y), y) \notin R$
parse σ as (r, s)	abort
$\hat{s} = s - y$	$\sigma \leftarrow \mathcal{O}_{SM2-Sign}(m)$
$Z = y(X + G)$	parse σ as (r, s)
$\pi_S \leftarrow S((G, X, Y, Z), 1)$	$\hat{s} = s - y$
$\hat{\sigma} = (r, s, Z, \pi_S)$	$Z = y(X + G)$
$\sigma^* \leftarrow \mathcal{A}^{\mathcal{O}_S(\cdot), \mathcal{O}_{PS}(\cdot)}(\hat{\sigma}, I_Y)$	$\pi_S \leftarrow S((G, X, Y, Z), 1)$
return (m^*, σ^*)	$\mathcal{Q} = \mathcal{Q} \cup \{m\}$
$\mathcal{O}_S(m)$	return (r, \hat{s}, Z, π_S)
$\sigma \leftarrow \mathcal{O}_{SM2-Sign}(m)$	
$\mathcal{Q} = \mathcal{Q} \cup \{m\}$	
return σ	

综上所述, 可知 $|\Pr[\text{Game}_0 = 1] - \Pr[\text{Game}_4 = 1]| \leq \text{negl}_1(\lambda) + \text{negl}_2(\lambda) + \text{negl}_3(\lambda) + \text{negl}_4(\lambda) \leq \text{negl}(\lambda)$. 即 $\Pr[\text{Game}_0 = 1] \leq \Pr[\text{Game}_4 = 1] + \text{negl}(\lambda)$. 因为 \mathcal{S} 可完美模拟 Game_4 , 所以 \mathcal{A} 在实验 aWitExt 中的优势为:

$$\mathbf{Adv}_{\mathcal{A}, \text{aWitExt}} = \Pr[\text{Game}_0 = 1] \leq \Pr[\text{Game}_4 = 1] + \text{negl}(\lambda) \leq \mathbf{Adv}_{\mathcal{S}, \text{sSigForge}} + \text{negl}(\lambda)$$

因此, 上述 SM2 适配器签名满足证据可提取性. \square

5 基于离线证明的 SM2 适配器签名

本节中, 我们对 SM2 适配器签名 SM2-ASv1 进一步优化, 基于离线证明技术设计更加高效的 SM2 适配器签名. 根据 $Z = (x + 1)Y = y(X + G)$, 可分别以 x 和 y 为证据证明 Z 的结构, 其中, 以 x 为证据, 该证明只能由预签名方生成; 以 y 为证据, 该证明可由困难关系生成方生成.

根据 SM2-ASv1, 该方案主要包含了两对困难关系 $((Y, \pi_Y), y)$ 和 $((G, X, Y, Z), x)$ 的零知识证明, 其中 $((Y, \pi_Y), y)$ 的证明可由困难关系生成方离线生成并证明, 后者在预签名算法阶段以预签名私钥 x 为证据, 证明离散对数相等的困难关系 $\{((G, X, Y, Z), x) \mid \exists x \in \mathbb{Z}_n, \text{ s.t. } X = xG \wedge Z - Y = xY\}$. 根据 SM2 签名算法特点, $((G, X, Y, Z), x)$ 的证明过程与预签名过程中的随机数选择以及签名消息是独立的, 该部分可由预签名方在离线阶段执行. 离线证明构造较为直观, 即预签名方在接收 (或生成) 困难关系 (Y, π_Y) 后, 先离线生成 Z 并进行离线证明, 后续对消息 m 进行预签名时, 再选择随机数并生成预签名值. 因此,

该方案可根据 SM2-ASv1 直接得到. 同时, 该过程依旧保持困难关系 $((Y, \pi_Y), y)$ 的自证明结构, 并不影响 SM2 适配器签名算法的安全性.

本节主要构造由困难关系生成方以 y 为证据离线证明 Z 结构的 SM2 适配器签名方案 SM2-ASv2. 令离散对数相等困难关系为 $R = \{(I = (G, I_Y = (Y, \pi_Y), X, Z, \pi), y) | \exists y \in \mathbb{Z}_n, \text{ s.t. } Y = yG \wedge Z - Y = yX \wedge V(I) \rightarrow 1\}$, 其中 P 和 V 分别表示零知识证明中的证明和验证算法. 基于 SM2 的适配器签名算法构造如下:

- $\text{pSign}((X, x), m, I) \rightarrow \hat{\sigma}$. 预签名算法输入公私钥对 (X, x) , 消息值 m 和困难实例 $I = (G, I_Y, X, Z, \pi)$, 可离线验证 I . 如果 $V(I) \rightarrow 0$, 输出 \perp , 否则计算 $e = H(m)$, 选择随机数 $k \leftarrow \mathbb{Z}_n$, 并计算 $\hat{K} = kG + Z$, $r_x = f(\hat{K})$, $r = e + r_x$, $\hat{s} = (1 + x)^{-1}(k - rx) \bmod n$, 最后输出预签名值 $\hat{\sigma} = (r, \hat{s})$.
- $\text{pVrfy}(X, m, I, \hat{\sigma}) \rightarrow 0/1$. 预签名验证算法输入公钥 X , 消息 m , 困难实例 $I = (G, I_Y, X, Z, \pi)$, 以及预签名值 $\hat{\sigma} = (r, \hat{s})$, 可离线验证 I , 如果 $V(I) \rightarrow 0$, 输出 \perp , 否则计算 $e = H(m)$, $\hat{K} = (\hat{s} + r)X + \hat{s}G + Z$, $r' = f(\hat{K}) + e \bmod n$. 如果 $r' = r$, 该算法输出 1, 否则输出 0.
- $\text{Adapt}(\hat{\sigma}, y) \rightarrow \sigma$. 适配算法输入预签名值 $\hat{\sigma} = (r, \hat{s})$ 和证据 y , 计算 $s = \hat{s} + y \bmod n$, 并输出签名值 $\sigma = (r, s)$.
- $\text{Ext}(\sigma, \hat{\sigma}, I) \rightarrow y$. 证据提取算法输入签名值 σ , 预签名值 $\hat{\sigma}$ 和困难实例 I , 计算 $y = s - \hat{s} \bmod n$. 如果 $(I, y) \in R$, 输出 y , 否则输出 \perp .

$\text{pSign}((X, x), m, I) \rightarrow \hat{\sigma}$	$\text{pVrfy}(X, m, I, \hat{\sigma}) \rightarrow 0/1$	$\text{Adapt}(\hat{\sigma}, y) \rightarrow \sigma$
if $V(I) \rightarrow 0$ outputs \perp . else, $e = H(m)$ $k \leftarrow \mathbb{Z}_n$ $r_x = f(kG + Z)$ $r = e + r_x$ $\hat{s} = (1 + x)^{-1}(k - rx) \bmod n$ return $\hat{\sigma}$	if $V(I) \rightarrow 0$ outputs \perp . else, $e = H(m')$ $R' = (\hat{s} + r)X + \hat{s}G + Z$ $r' = f(R') + e \bmod n$ If $r' = r$, output 1, else, output 0. return 0/1	$s = \hat{s} + y \bmod n$ return σ $\text{Ext}(\sigma, \hat{\sigma}, I) \rightarrow y$ $y = s - \hat{s} \bmod n$ If $((Y, Z)y) \in R$, output y , else outputs \perp return y/\perp

图 2 基于离线证明的 SM2 适配器签名

Figure 2 SM2-based adaptor signature scheme with offline proof

本节主要构造的 SM2 适配器签名, 如图 2 所示, 困难实例 I 包含 $Y = yG$ 和 $Z = y(X + G)$, 皆以 y 为证据, 具体的零知识证明可由困难关系生成方生成. 因此, 预签名方接收到该困难实例 I 可直接离线验证其正确性, 随后需要对消息进行预签名时, 根据离线验证的 Y, Z , 并选择随机数对消息进行预签名. 该方法避免了在线预签名阶段中计算 Z 的过程, 并可通过零知识证明其结构, 与在线 SM2-ASv1 和现有的 ECDSA-AS 相比较, SM2-ASv2 在线预签名的计算效率更高. 由于 Z 的零知识证明加入到离线证明的困难关系中, 预签名中不需要过多的包含 Z 的信息. 因此, SM2-ASv2 的签名值更小. 同样, 根据对 SM2-ASv2 的分析, 离线证明过程依旧保证自证明结构 $((Y, \pi_Y), y)$, 不影响适配器签名的安全性, 具体分析可参考 SM2-ASv1.

6 方案效率和实验结果

6.1 理论分析

本节分析现有基于 ECDSA 的适配器签名和基于 SM2 的适配器签名, 分别在通信成本和计算效率两个方面进行对比, 具体结果可见表 1. 在进行计算效率对比时, 我们主要考虑了较为复杂的点乘运算, 忽略了其它较为简单的运算, 如点加运算等. Moreno-Sanchez 等 [27] 提出的首个 ECDSA-AS 并未给出可证明安全. 随后 Aumayr 等 [5] 在 [27] 的基础上, 提出自证明结构 $\{(I_Y = (Y, \pi_Y), y) | Y = yG \wedge V(Y, \pi_Y) \rightarrow 1\}$, 并给出了可证明安全的 ECDSA-AS. 但是该方案在预签名过程中, 以签名随机数为证据, 证明满足离散对数相等的困难关系 $\{((G, \hat{K}, Y, K), k) | \exists k \in \mathbb{Z}_n, \text{ s.t. } \hat{K} = kG \wedge K = kY\}$. 一方面该部分只能由签名方生成 (预签名随机数由签名方生成), 无法由其他方离线生成, 需要 4 次点乘操作. 另一方面每次重新进行预

签名, 需要选择新的随机数 (不同的预签名值需要新的随机数), 也需要重新进行零知识证明. 因此, 该方案的预签名过程更加复杂, 应用不够灵活.

我们沿用现有自证明结构 [5] 设计了基于 SM2 的适配器签名 SM2-ASv1. 该方案在预签名过程中以签名私钥为证据, 需要证明满足离散对数相等的困难关系 $\{((G, X, Y, Z), x) | \exists x \in \mathbb{Z}_n, \text{ s.t. } X = xG \wedge Z - Y = xY\}$. 基于 SM2 签名算法的结构特性, 我们设计了支持离线证明的 SM2 适配器签名 SM2-ASv2. 一方面, SM2-ASv1 中关于 Z 结构的零知识证明可由签名方拿到困难实例 I_Y 后离线进行, 提升在线预签名操作的效率; 另一方面, 还能以 y 为证据, 证明 Z 的结构 $\{((G, Y, X, Z), y) | \exists y \in \mathbb{Z}_n, \text{ s.t. } Y = yG \wedge Z - G = yX\}$. 该证明与预签名随机数和消息无关, 对于一个困难关系可以进行多次预签名, 而且该部分证明可由困难关系生成方离线证明. 因此, SM2-ASv2 避免了在线签名阶段对于 Z 结构的零知识证明, 只需要 1 次点乘操作, 较大的提升了在线预签名的效率.

表 1 通信成本与效率对比
Table 1 Communication cost and efficiency comparison

方案	预签名验证 公钥尺寸	预签名 私钥尺寸	预签名 尺寸	预签名 计算 (在线)	预签名 验证	是否可 证明安全
ECDSA-AS [27]	$ G $	$ \mathbb{Z}_n $	$4 \mathbb{Z}_n + G $	4Exp	6Exp	?
ECDSA-AS [5]	$ G $	$ \mathbb{Z}_n $	$4 \mathbb{Z}_n + G $	4Exp	6Exp	✓
SM2-ASv1	$ G $	$ \mathbb{Z}_n $	$4 \mathbb{Z}_n + G $	4Exp	6Exp	✓
SM2-ASv2	$ G $	$ \mathbb{Z}_n $	$2 \mathbb{Z}_n $	Exp	6Exp	✓

[‡] 其中, $|G|$ 表示椭圆曲线上点的尺寸, $|\mathbb{Z}_n|$ 表示 \mathbb{Z}_n 群元素的尺寸, Exp 表示椭圆曲线上点乘运算, ? 表示未知.

6.2 实验分析

为了评估适配器签名方案的实际性能, 我们基于 OpenSSL 库分别实现了 ECDSA-AS [5], SM2-ASv1 和 SM2-ASv2. 执行环境为: Intel Core i5 CPU 2.3 GHz, 8GB RAM, macOS High Sierra 10.13.3 system.

我们主要对比了三种方案在线预签名操作的计算效率以及预签名验证算法的计算效率, 具体如图 3 所示. 我们分别运行各方案的程序 1000 次, 发现 ECDSA-AS [5], SM2-ASv1 和 SM2-ASv2 的在线预签名操作单次平均耗时分别为 $166.54\mu s$, $176.27\mu s$, $37.14\mu s$; 预签名验证操作单次平均耗时分别为 $252.86\mu s$, $238.46\mu s$, $234.80\mu s$. 实验结果符合上述理论分析, SM2-ASv2 的在线预签名耗时仅为 ECDSA-AS 的 1/4.

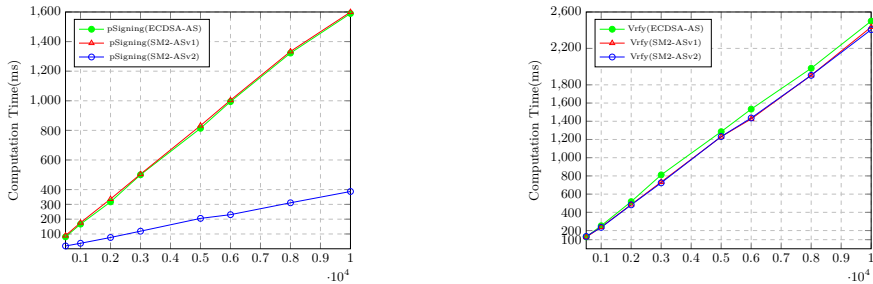


图 3 (a) 预签名算法和 (b) 验证算法效率对比

Figure 3 Efficiency comparison of (a) pre-signing and (b) verification algorithm

7 应用

在本节中, 我们根据现有适配器签名应用 [16, 13, 26, 5], 介绍 SM2 适配器签名的两个典型区块链应用: 原子交换协议和支付通道网络.

7.1 原子交换协议

原子交换协议 (atomic swap) 能在无可信第三方的情况下公平地实现不同密码货币之间的交换. 用户 U_1 和 U_2 想要公平地交换不同的密码货币 c_1 和 c_2 , 其中公平性体现在双方完成交换或者双方都交换失

败. 最初, 原子交换协议 [15] 基于哈希函数和时间锁 (timelock) 设计, 可实现拥有哈希函数原像才能提取货币的功能, 时间锁保证后提取货币的用户有足够的时间进行货币提取, 但该方法要求密码货币的两种脚本语言都支持哈希原像条件脚本 (preimage conditioned scripts). 随后, Poelstra [16] 将哈希条件嵌入到签名算法中, 给出了该协议的无脚本版本. 因此, 基于适配器签名设计原子交换协议框架 [15, 16, 26], 在现有基于 SM2 签名算法的区块链应用中, 可使用 SM2 的适配器签名构造原子交换协议, 具体如图 4 所示:

- 协议建立阶段: U_1 运行 $\text{GenR}(1^\lambda) \rightarrow (Y, y)$ 生成困难关系 (Y, y) , 计算零知识证明 $P(Y, y) \rightarrow \pi$, 令 $I = (Y, \pi)$. 生成关于传输货币 c_1 给 U_2 的预交易 tx_1 , 并生成预签名 $\hat{\sigma}_1 \leftarrow \text{pSign}((X_1, x_1), tx_1, I)$; 将 $I, \hat{\sigma}_1, tx_1$ 发送给 U_2 ; U_2 可验证困难实例 I 和预签名值 $\hat{\sigma}_1$, 如果 $V(I) \rightarrow 0$ 或者 $\text{pVrfy}(X_1, tx_1, I, \hat{\sigma}_1) \rightarrow 0$, 拒绝交易; 否则生成关于传输货币 c_2 给 U_1 的预交易 tx_2 , 并生成预签名 $\hat{\sigma}_2 \leftarrow \text{pSign}((X_2, x_2), tx_2, I)$; 将 $\hat{\sigma}_2, tx_2$ 发送给 U_1 ;
- 交换阶段: U_1 验证预签名值, 如果 $\text{pVrfy}(X_2, tx_2, I, \hat{\sigma}_2) \rightarrow 0$, 则拒绝交易; 否则根据证据 y , 运行适配算法 $\sigma_2 \leftarrow \text{Adapt}(\hat{\sigma}_2, y)$ 将 U_2 的预签名适配成 SM2 签名, 即获得 U_2 传输货币 c_2 给 U_1 的 SM2 签名值 σ_2 , 在链上公布 σ_2 可获得 c_2 ; U_2 获得签名值 σ_2 , 运行提取算法获得证据 $y \leftarrow \text{Ext}(\sigma_2, \hat{\sigma}_2, I)$, 并运行适配算法 $\sigma_1 \leftarrow \text{Adapt}(\hat{\sigma}_1, y)$, 获得 U_1 传输货币 c_1 给 U_2 的 SM2 签名值 σ_1 , 在链上公布 σ_1 获得 c_1 .

$U_1((X_1, x_1), X_2, c_1)$		$U_2((X_2, x_2), X_1, c_2)$	
$\text{GenR}(1^\lambda) \rightarrow (Y, y), P(Y, y) \rightarrow \pi$			
$I = (Y, \pi), \hat{\sigma}_1 \leftarrow \text{pSign}((X_1, x_1), I, tx_1)$		$\xrightarrow{I, \hat{\sigma}_1, tx_1}$	If $V(I) \rightarrow 0$ or $\text{pVrfy}(X_1, tx_1, I, \hat{\sigma}_1) \rightarrow 0$ Output \perp
If $\text{pVrfy}(X_2, tx_2, I, \hat{\sigma}_2) \rightarrow 0$ Output \perp		$\xleftarrow{\hat{\sigma}_2, tx_2}$	else, $\hat{\sigma}_2 \leftarrow \text{pSign}((X_2, x_2), tx_2, I)$
eLse, $\sigma_2 \leftarrow \text{Adapt}(\hat{\sigma}_2, y)$ Publish σ_2 on blockchain		$\xrightarrow{\sigma_2}$	$y' \leftarrow \text{Ext}(\sigma_2, \hat{\sigma}_2, I)$ $\sigma_1 \leftarrow \text{Adapt}(\hat{\sigma}_1, y)$ Publish σ_1 on blockchain

图 4 基于 SM2 适配器签名构造原子交换协议

Figure 4 Atomic swap protocol based on SM2-based adaptor signature

7.2 支付通道网络

支付通道网络 (payment channel network, PCN) [11, 8, 7, 13, 5] 是目前解决区块链可扩展性差, 吞吐量低的主流方案. 具体地说, 区块链上每笔交易需要各方 (矿工) 认可和存储, 导致交易吞吐量较差, 但 PCN 可通过将部分交易移到链下来提高吞吐量. 在 PCN 中, 交易双方可将货币锁定在一个通道中, 只要有足够的余额, 就可以进行即时和任意多次的交易. 目前, 最受欢迎的 PCN 应用是基于比特币的闪电网络 [8]. PCN 允许各方进行多跳支付, 即没有直接通道的参与方可以使用中介节点的通道来实现支付. 在多跳支付中, 各方需要同步路线上每个通道信息, 要求所有通道都同步更新, 或者都不更新. 闪电网络通过使用哈希锁合约 (hash-time lock contract, HTLC) 实现上述要求. 然而, Malavolta 等 [13] 提出虫洞攻击可破坏 HTLC 机制的安全性, 并基于 ECDSA 适配器签名构造匿名多跳锁 (anonymous multi-hop lock, AMHL) 构建了安全的 PCN.

在现有基于 SM2 签名算法的区块链应用中, 我们可使用 SM2 的适配器签名, 采用 Malavolta 等 [13, 26] 的构造框架, 实现 PCN 的多跳支付. 具体应用中, 发送者 S (或 U_0) 通过中介节点 U_1, \dots, U_{k-1} 向接收方 R (或 U_k) 进行支付, 为了方便, 我们省略中介节点的费用, 令 (X_j, x_j) 是 U_j 的 SM2 签名公私钥对, G 是阶为 n 的基点, tx_j 是指 U_j 到 U_{j+1} 交易. 具体协议如图 5:

- 建立阶段: S 选择随机数 $l_j \in \mathbb{Z}_n, j = 0, \dots, k-1$, 计算 $y_j = \sum_{i=0}^j l_i \bmod n$ 和 $Y_j = y_j G$, 并证明 $\pi_{Y_j} \leftarrow P(Y_j, y_j)$, 计算 $Z_j = y_j X_j + Y_j, I_j = (G, (Y_j, \pi_{Y_j}), X_j, Z_j), \pi_j \leftarrow P(I_j), j = 0, \dots, k-1$.

S 发送 $(Y_{j-1}, Y_j, Z_j, \pi_j, l_j)$ 给 U_j , 发送 (Y_{k-1}, y_{k-1}) 给 R . U_j 可验证传输值的正确性, $V(I_j) \rightarrow 1$ 和 $Y_j - Y_{j-1} = l_j G$, 否则拒绝交易. R 可验证 $Y_{k-1} = y_{k-1} G$, 否则拒绝交易.

- 支付阶段: S 计算预签名 $\hat{\sigma}_0 \leftarrow \text{pSign}((X_0, x_0), tx_0, I_0)$ 并发送给 U_1 , U_1 验证预签名值 $\text{pVrfy}(X_0, tx_0, I_0, \hat{\sigma}_0) \rightarrow 1$, 否则拒绝交易. U_j 从 U_{j-1} 收到预签名 $\hat{\sigma}_{j-1}$, 并验证正确后, 计算预签名 $\hat{\sigma}_j \leftarrow \text{pSign}((X_j, x_j), tx_j, I_j)$, $j = 1, \dots, k-1$ 发送给 U_{j+1} . 所有预签名传输完成, 即 R 获得预签名 $\hat{\sigma}_{k-1}$, U_j 获得预签名 $\hat{\sigma}_{j-1}$, 并计算了预签名值 $\hat{\sigma}_j$. R 根据证据 y_{k-1} 可将预签名值 $\hat{\sigma}_{k-1}$ 适配成 SM2 签名值 $\sigma_{k-1} \leftarrow \text{Adapt}(\hat{\sigma}_{k-1}, y_{k-1})$ 获得 U_{k-1} 的支付. R 发送 σ_{k-1} 给 U_{k-1} ; U_{k-1} 可验证 σ_{k-1} 的正确性, $\text{Vrfy}(X_{k-1}, tx_{k-1}, \sigma_{k-1}) \rightarrow 1$, 否则拒绝交易, 并运行提取算法提取证据 $y_{k-1} \leftarrow \text{Ext}(\sigma_{k-1}, \hat{\sigma}_{k-1}, I_{k-1})$, 计算 $y_{k-2} = y_{k-1} - l_{k-1}$, 运行适配算法计算 SM2 签名值 $\sigma_{k-2} \leftarrow \text{Adapt}(\hat{\sigma}_{k-2}, y_{k-2})$ 获得 U_{k-2} 的支付, 并将 σ_{k-2} 发送给 U_{k-2} . 如此, U_j 获得 SM2 签名 σ_j 可通过提取算法获得 $y_j \leftarrow \text{Ext}(\sigma_j, \hat{\sigma}_j, I_j)$, 并计算出 $y_{j-1} = y_j - l_j$, 然后通过适配算法计算 SM2 签名 $\sigma_{j-1} \leftarrow \text{Adapt}(\hat{\sigma}_{j-1}, y_{j-1})$ 获得 U_{j-1} 的支付, 并将 σ_{j-1} 发送给 U_{j-1} , $j = k-1, k-2, \dots, 1$. 最后, U_1 可计算 SM2 签名 σ_0 获得 S 的支付.

$S(x_0, X_j), j = 0, 1, \dots, k-1$	Setup	$U_j(x_j, X_j), j = 1, \dots, k$
$l_j \leftarrow_r \mathbb{Z}_n, j = 0, \dots, k-1$		
$y_j = \sum_{i=0}^j l_i \bmod n$		
$Y_j = y_j G, \pi_{Y_j} \leftarrow P(Y_j, y_j)$		
$Z_j = y_j X_j + Y_j$		
$I_j = (G, (Y_j, \pi_{Y_j}), X_j, Z_j)$	$\frac{Y_{j-1}, Y_j, Z_j, \pi_j, l_j}{Y_{k-1}, y_{k-1}} \rightarrow U_j$	If $V(I_j) \rightarrow 0$ or $Y_j - Y_{j-1} \neq l_j G$, U_j aborts.
$\pi_j \leftarrow P(I_j), j = 0, \dots, k-1$	$\rightarrow R$	If $Y_{k-1} \neq y_{k-1} G$, R aborts.
Payment		
$S \xrightarrow{\hat{\sigma}_0 \leftarrow \text{pSign}((X_0, x_0), tx_0, I_0)} U_1 \xrightarrow{\dots} \dots \xrightarrow{\hat{\sigma}_{j-1}} U_j \xrightarrow{\hat{\sigma}_j \leftarrow \text{pSign}((X_j, x_j), tx_j, I_j)} U_{j+1} \xrightarrow{\hat{\sigma}_{j+1}} \dots \xrightarrow{\dots} U_{k-1} \xrightarrow{\hat{\sigma}_{k-1} \leftarrow \text{pSign}((X_{k-1}, x_{k-1}), tx_{k-1}, I_{k-1})} R$		
$S \xleftarrow{\sigma_0 \leftarrow \text{Adapt}(\hat{\sigma}_0, y_0)} U_1 \xleftarrow{\dots} \dots \xleftarrow{\sigma_{j-1}} U_j \xleftarrow{\sigma_j \leftarrow \text{Adapt}(\hat{\sigma}_j, y_j)} U_{j+1} \xleftarrow{\sigma_{j+1}} \dots \xleftarrow{\dots} U_{k-1} \xleftarrow{\sigma_{k-1} \leftarrow \text{Adapt}(\hat{\sigma}_{k-1}, y_{k-1})} R$		

图 5 基于 SM2 适配器签名的多跳支付协议

Figure 5 Multi-hop payments based on SM2-based adaptor signature

8 总结

针对现有区块链技术的可扩展性差、交易吞吐量低等问题, 同时秉承积极推广国密算法应用的理念, 填补现有 SM2 适配器签名的空白, 我们基于“自证明结构”设计可证明安全的 SM2 适配器签名方案, 并且根据 SM2 签名的结构, 采用“离线证明技术”, 在不影响可证明安全性的情况下, 构造更加高效的 SM2 适配器签名方案. 该方案与现有 ECDSA 适配器签名方案相比, 在线预签名过程不需要额外的零知识证明, 计算效率更高, 应用优势更广. 最后, 我们基于适配器签名的现有应用, 分别给出 SM2 适配器签名在区块链上的两个具体应用, 为后续 SM2 签名算法的应用推广提供参考.

References

- [1] Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2008), <https://nakamotoinstitute.org/bitcoin/>
- [2] Bano, S., Sonnino, A., Al-Bassam, M., Azouvi, S., McCorry, P., Meiklejohn, S., Danezis, G.: Sok: Consensus in the age of blockchains. In: Proceedings of the 1st ACM Conference on Advances in Financial Technologies, AFT 2019, Zurich, Switzerland, October 21-23, 2019. pp. 183–198. ACM (2019), <https://doi.org/10.1145/3318041.3355458>
- [3] Gudgeon, L., Moreno-Sanchez, P., Roos, S., McCorry, P., Gervais, A.: Sok: Off the chain transactions. IACR Cryptol. ePrint Arch. 2019, 360 (2019), <https://eprint.iacr.org/2019/360>
- [4] Zamyatin, A., Al-Bassam, M., Zindros, D., Kokoris-Kogias, E., Moreno-Sanchez, P., Kiayias, A., Knottenbelt, W.J.: Sok: Communication across distributed ledgers. IACR Cryptol. ePrint Arch. 2019, 1128 (2019), <https://eprint.iacr.org/2019/1128>
- [5] Aumayr, L., Ersoy, O., Erwig, A., Faust, S., Hostáková, K., Maffei, M., Moreno-Sanchez, P., Riahi, S.: Generalized bitcoin-compatible channels. IACR Cryptol. ePrint Arch. 2020, 476 (2020), <https://eprint.iacr.org/2020/476>
- [6] Shan J Y, Gao S. Research progress on theory of blockchains[J]. Journal of Cryptologic Research, 2018, 5(5). 单进勇, 高胜. 区块链理论研究进展 [J]. 密码学报, 2018, 5(5).

- [7] Bitcoin Wiki: Payment channels (2018), <https://en.bitcoin.it/wiki/Paymentchannels>
- [8] J. Poon and T. Dryja: The bitcoin lightning network: Scalable off-chain instant payments (2016), <https://lightning.network/lightning-network-paper.pdf>
- [9] Update from the Raiden team on development progress, announcement of raidEX. <https://tinyurl.com/z2snp9e>. Feb.2017.
- [10] A. Poelstra. Lightning in Scriptless Scripts. mumblewimble team mailing list. <https://lists.launchpad.net/mumblewimble/msg00086.html>.
- [11] Decker, C., Wattenhofer, R.: A fast and scalable payment network with bitcoin duplex micropayment channels. In: Pelc, A., Schwarzmann, A.A. (eds.) Stabilization, Safety, and Security of Distributed Systems - 17th International Symposium, SSS 2015, Edmonton, AB, Canada, August 18-21, 2015, Proceedings. Lecture Notes in Computer Science, vol. 9212, pp. 3–18. Springer (2015), https://doi.org/10.1007/978-3-319-21741-3_1
- [12] Ekey, L., Faust, S., Hostáková, K., Roos, S.: Splitting payments locally while routing interdimensionally. IACR Cryptol. ePrint Arch. 2020, 555 (2020), <https://eprint.iacr.org/2020/555>
- [13] Malavolta, G., Moreno-Sanchez, P., Schneidewind, C., Kate, A., Maffei, M.: Anonymous multi-hop locks for blockchain scalability and interoperability. In NDSS 2019, <https://www.ndss-symposium.org/ndss-paper/anonymous-multi-hop-locks-for-blockchain-scalability-and-interoperability/>
- [14] Miller, A., Bentov, I., Bakshi, S., Kumaresan, R., McCorry, P.: Sprites and state channels: Payment networks that go faster than lightning. In: Goldberg, I., Moore, T. (eds.) Financial Cryptography and Data Security - 23rd International Conference, FC 2019. Lecture Notes in Computer Science, vol. 11598, pp. 508–526. Springer (2019), https://doi.org/10.1007/978-3-030-32101-7_30
- [15] Nolan, T.: Alt chains and atomic transfers, <https://bitcointalk.org/index.php?topic=193281.msg2224949#msg2224949>
- [16] Poelstra, A.: Adaptor signatures and atomic swaps from scriptless scripts, <https://github.com/ElementsProject/scriptless-scripts/tree/master/slides/2017-05-milan-meetup>
- [17] Deshpande, A., Herlihy, M.: Privacy-preserving cross-chain atomic swaps. In: Bernhard, M., Bracciali, A., Camp, L.J., Matsuo, S., Maurushat, A., Rønne, P.B., Sala, M. (eds.) Financial Cryptography and Data Security - FC 2020, https://doi.org/10.1007/978-3-030-54455-3_38
- [18] Guger, J.: Bitcoin-monero cross-chain atomic swap. IACR Cryptol. ePrint Arch. 2020, 1126 (2020), <https://eprint.iacr.org/2020/1126>
- [19] GM/T 0003-2012, Public Key Cryptographic Algorithm SM2 Based on Elliptic Curves. 2010 <http://www.gmbz.org.cn/main/viewfile/20180108015515787986.html>
GM/T 0003-2012, SM2 椭圆曲线公钥密码算法.2010.
- [20] Wang Z H, Zhang Z F. Overview on Public Key Cryptographic Algorithm SM2 Based on Elliptic Curves. Journal of Information Security Research, 2016(11).
汪朝晖, 张振峰. SM2 椭圆曲线公钥密码算法综述 [J]. 信息安全研究, 2016(11).
- [21] Schnorr, C.: Efficient identification and signatures for smart cards. In: Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings. pp. 239–252 (1989), https://doi.org/10.1007/0-387-34805-0_22
- [22] Erwig, A., Faust, S., Hostáková, K., Maitra, M. Riahi, S.: Two-Party Adaptor Signatures From Identification Schemes. Public-Key Cryptography, 2021.
- [23] American National Standards Institute: X9.62: Public key cryptography for the financial services industry: The elliptic curve digital signature algorithm (ecdsa) (2005)
- [24] Hoffman, P.E., Wijngaards, W.C.A.: Elliptic curve digital signature algorithm (DSA) for DNSSEC. RFC 6605, 1–8 (2012), <https://doi.org/10.17487/RFC6605>
- [25] Ducas, L., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: Crystals – Dilithium: Digital signatures from module lattices. In: CHES. vol. 2018-1 (2018)
- [26] Esgin, M.F., Ersoy, O., Erkin, Z.: Post-quantum adaptor signatures and payment channel networks. In: Chen, L., Li, N., Liang, K., Schneider, S.A. (eds.) Computer Security - ESORICS 2020, Lecture Notes in Computer Science, vol. 12309, pp. 378–397. Springer (2020), https://doi.org/10.1007/978-3-030-59013-0_19
- [27] Moreno-Sanchez, P., Kate, A.. Scriptless Scripts with ECDSA. lightning-dev mailing list. <https://lists.linuxfoundation.org/pipermail/lightning-dev/attachments/20180426/fe978423/attachment-0001.pdf>
- [28] Fischlin, M.: Communication-efficient non-interactive proofs of knowledge with online extractors. In: Shoup, V. (ed.) Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3621, pp. 152–168. Springer (2005), https://doi.org/10.1007/11535218_10

- [29] Zhang, Z., Yang, K., Zhang, J., Chen, C.: Security of the SM2 signature scheme against generalized key substitution attacks. In: Chen, L., Matsuo, S. (eds.) Security Standardisation Research - Second International Conference, SSR 2015, Tokyo, Japan, December 15-16, 2015, Proceedings. Lecture Notes in Computer Science, vol. 9497, pp. 140–153. Springer (2015), https://doi.org/10.1007/978-3-319-27152-1_7