密码学报 ISSN 2095-7025 CN 10-1195/TN
Journal of Cryptologic Research, 2017, 12(27): 1001–1013

E-mail: jcr@cacrnet.org.cn
http://www.jcr.cacrnet.org.cn
Tel/Fax: +86-10-82789618

支持离线批量证明的 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 (如密码货币) 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: (1) 降低链上成本, (2) 提高交易的可替代性, (3) 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, 我们基于国密 SM2 签名构造出适配器签名方案, 并在随机预言机模型下给出安全性证明. 随后, 我们根据 SM2 签名的结构特点, 提出离线批量证明技术对 SM2 适配器签名进行优化, 获得了可支持离线批量证明的 SM2 适配器签名方案. 该方案以困难关系的证据作为连系 SM2 签名验证公钥和预签名公开参数的证据, 使得 SM2 适配器签名在预签名阶段的零知识证明可由困难关系选择方离线批量生成, 与现有 ECDSA 适配器签名相比更加高效, 更加灵活, 更适用于区块链应用. 最后, 我们给出 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; 批量证明; 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

The SM2-based Adaptor Signature with Offline Batch Proof and Its Applications

Abstract: Adaptor signature, also known as scriptless script, 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 (1) low on-chain cost, (2) improved fungibility of transactions, (3) 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 design an adaptor signature based on SM2 signature algorithm, and prove security under the random oracle model. Then, based on the structure of SM2 signature, we develop offline batch proof technique to optimize it and get a SM2-based adaptor signature supporting offline batch proof. In particular, this scheme uses the witness of hard relation as the witness of fixed SM2 verification key and pre-signing public parameter, so that the zero-knowledge proof uesd in the pre-signing algorithm of SM2-based adaptor signature can be generated in offline and batch. Compared with the existing ECDSA-based adaptor signature, the scheme is more efficient, flexible and suitable for blockchain application. Finally, we show two applications of SM2-based adapter signature in blockchain: atomic swap protocol and payment channel network.

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

收稿日期: 2016-05 定稿日期: 2017-09

1 引言

自 2009 年比特币 [1] 出现, 其底层的区块链结构引起了广泛关注. 区块链巧妙地融合密码、共识机制、P2P 网络等技术, 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 (如密码货币) 也面临着可扩展性差、吞度量低等应用问题 [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, 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. 但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, 实现用户之间通过该通道进行任意次数的链下交易, 是目前较为高效且部署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) 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: (1) 拥有签名私钥的签名者才能生成预签名; (2) 拥有困难关系的证据才能将预签名值转化为签名值; (3) 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, 比如原子交换协议中, 双方可基于适配器签名实现跨链的公平交换. 发起方 (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 $\hat{\sigma}_1$ 发送给交换方; 交换方根据实例 Y 也对交换交易进行预签名, 并将预签名值 $\hat{\sigma}_2$ 返回; 发起方拥有困难关系的证据 y 可将预签名值 $\hat{\sigma}_2$ 转换成获得交换的签名值 σ_2, 在链上公布 σ_2 可获得交换的货币; 交换方根据 σ_2 和 $\hat{\sigma}_2$ 可提取证据 y, 并将 $\hat{\sigma}_1$ 转换成获得交换的签名值 σ_1, 在链上公布 σ_1 可获得交换的货币. 至此, 双方完成公平交换交易. 适配器签名通过将签名授权和证据提取巧妙地结合, 完成公平交换的复杂操作, 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有重要的应用价值, 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发布的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, 25], 以及格签名 [26, 27] 等方案构造, 基于 SM2 签名构造适配器签名的研究较少, 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] 在预签名阶段需要计算预签名公开参数和对应的零知识证明, 导致预签名操作计算效率较低. 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.

1.1 本文贡献

在本文中, 我们首先基于国密 SM2 签名算法构造适配器签名方案 (**SM2-AS1**), 并在随机预言机模型 (random oracle model, ROM) 下基于 SM2 签名的安全性给出安全性证明. 然后, 我们根据 SM2 签名结构, 提出离线批量证明技术对 SM2 适配器签名进一步优化. 优化后的 SM2 适配器签名 (**SM2-AS2**) 以困难关系的证据作为连系 SM2 签名验证公钥和预签名公开参数的证据, 能实现所有预签名参与方的预签名公开参数和对应的零知识证明由困难关系选择方 (拥有困难关系的证据) 离线批量生成, 与现有 ECDSA 适配器签名方案 (ECDSA-AS) [5] 相比, 在线预签名操作的计算效率更高, 预签名值尺寸更小. 随后, 我们对 SM2-AS1, SM2-AS2 和 ECDSA-AS 进行性能评估和对比, 发现 SM2-AS2 的在线预签名操作需要较少的点乘运算, 耗时仅为 ECDSA-AS 的 1/4. 最后, 我们给出 SM2 适配器签名在区块链中的具体应用, 发现 SM2-AS2 比 ECDSA-AS 在 (批量) 原子交换协议和支付通道网络的应用中, 更加灵活高效. 主要技术介绍如下:

SM2 适配器签名. SM2 适配器签名是 SM2 签名对离散对数困难关系的扩展, 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. 其中, 预签名算法根据签名私钥和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. 对 SM2-AS1 介绍如下: 令 (X,x) 为 SM2 签名算法的公私钥对, $(I_Y=(Y,\pi_Y),y)\in R$ 是离散对数困难关系 [5]. 计算预签名值: 计算预签名公开参数 $Z=(x+1)Y$, 以 x 为证据证明 X 和 $Z-Y$ 具有相同的离散对数关系 $(Z-Y=xY, X=xG)$: $\pi_Z\leftarrow \mathsf{P}_Z((G,X,Y,Z-Y),x)$, 选择随机数 k 计算 $\hat{K}=kG+Z=(k+y(x+1))G=(r_x,r_y)$, $r=r_x+H(m) \bmod n$, $\hat{s}=(1+x)^{-1}(k-r\cdot x) \bmod n$, 即预签名值 $\hat{\sigma}=(r,\hat{s},Z,\pi_Z)$; 适配算法可基于证据 y 将 $\hat{\sigma}$ 转化为完整 SM2 签名值 $\sigma=(r,s)$, 其中 $s=\hat{s}+y=(x+1)^{-1}(k+y(x+1)-r\cdot x)$; 提取算法可根据预签名值 $\hat{\sigma}$ 和签名值 σ 提取证据 $y=s-\hat{s}$.

安全性证明. 适配器签名需要满足不可伪造性和证据可提取性, 其中不可伪造性与签名的安全性区别在于敌手可额外访问预签名预言机, 同时在证据可提取性的定义中, 敌手在挑战阶段能选择挑战困难关系访问预签名预言机. SM2 适配器签名采用具有直线证据提取性的非交互零知识的知识证明 [28] 提供的 "自证明结构" [5] 保证可证明安全. 具体而言, 在原离散对数困难关系 (Y,y) 的基础上增加额外证明, 将其扩展为具有 "自证明结构" 的离散对数困难关系 $(I_Y=(Y,\pi_Y),y)\in R$ [5], 其中 $\pi_Y\leftarrow \mathsf{P}_Y(Y,y)$ 证明存在证据 y 使得 $Y=y\cdot G$.

在安全性证明中, 模拟器可根据自己的随机预言机和 SM2 签名预言机模拟敌手要访问的这两个预言机, 并根据困难关系的 "自证明结构", 在随机预言机模型下通过直线证据提取器的功能提取证据 y [1], 然后结合 SM2 签名预言机输出的签名值计算预签名值, 模拟 SM2 适配器签名的预签名预言机. 此外, 对于证据可提取性的证明, 即使在挑战阶段, 敌手可自适应选择挑战困难关系, 但是模拟器依旧可根据 "自证明结构" 提取证据 y, 并模拟敌手的预签名预言机. 因此, 模拟器可模拟敌手的视角, 将 SM2 适配器签名的安全性归约到 SM2 签名的安全性上.

离线批量证明. SM2 适配器签名和现有 ECDSA 适配器签名 ECDSA-AS [5] 的预签名阶段, 都需要计算预签名公开参数和对应的零知识证明. 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. 具体而言, 在预签名过程中, ECDSA-AS [5] 需要以预签名的随机数 k 作为证据, 计算预签名公开参数 $K=kY$, 并证明离散对数相等的困难关系 $\{((G,\hat{K},Y,K),k)|\exists\ k\in\mathbb{Z}_n,\ \text{s.t. } \hat{K}=kG\wedge K=kY\}$. 因此, 该证明只能由预签名方生成 (随机数由预签名方自己选择), 对于相同的困难关系 (I_Y,y), 多次预签名计算需要选用不同的随机数, 且对于新的随机数需要进行新的零知识证明, 在批量原子交换协议应用中计算效率较低.

SM2-AS1 适配器签名方案在预签过程中以签名私钥 x 作为证据, 计算预签名公开参数 $Z=xY+Y$, 并证明离散对数相等的困难关系 $\{((G,X,Y,Z),x)|\exists\ x\in\mathbb{Z}_n,\ \text{s.t. } X=xG\wedge Z-Y=xY\}$ [2]. 该证明也只能由预签名方生成, 但是相较于 ECDSA-AS 方案, 对于相同的困难关系 (I_Y,y) 对应相同的预签名公开参数和零知识证明, 在批量原子交换协议中计算效率较高.

根据 SM2 签名和预签名公开参数的结构, 我们提出离线批量证明技术构造更加高效的可支持离线批量证明的适配器签名 SM2-AS2. 该方案使用困难关系的证据 y 为证据, 计算预签名公开参数 $Z=y(X+G)$, 证明离散对数相等的困难关系 $\{((G,Y,X+G,Z),y)|\exists\ y\in\mathbb{Z}_n,\ \text{s.t. } Z=y(X+G)\wedge Y=yG\}$. 该证明可由困难关系选择方 (拥有 y) 生成 $\pi_Z\leftarrow \mathsf{P}_Z((G,Y,X+G,Z),y)$. 在具体应用中, 困难关系选择方可以在协议执行前, 对需要进行预签名的所有参与方 U_i, 离线批量生成关于预签名公开参数 Z_i 和对应的零知识证明 π_{Z_i}, 该操作可避免大量参与方在预签名阶段自己计算, 能极大地提升参与方预签名的计算效率. 当前的 ECDSA-AS [5] 和 SM2-AS1, 预签名阶段的零知识证明只能在接收到实例 (Y,π_y) 后, 由参与方各自以随机数 k (ECDSA-AS) 或签名私钥 x (SM2-AS1) 为证据独立地生成预签名公开参数和对应

[1] 该零知识证明系统需要满足直线证据提取性 [28]. 在安全性证明中, 模拟器需要使用直线提取器 (straight-line extractor), 有时也叫在线提取器 (online extractor) 提取证据 y, 不能通过重绕提取器 (rewinding extractor) 来提取证据.

[2] 该零知识证明系统不需要满足较强的直线证据提取性, 可使用高效的 Fiat-Shamir 启发式转换 [29] Chaum-Pedersen 的 $\sum$-协议 [30] 实现.

的零知识证明, 难以批量生成. 支持批量证明的 SM2-AS2 更加适用于批量原子交换协议和中介节点较多的支付通道网络, 能较大幅度的提高参与方在预签名阶段的计算效率.

SM2-AS2 采用离线批量证明技术只是预签名公开参数 Z 和零知识证明 π_Z 的生成方式不同, 并不影响困难关系 (I_Y, y) 的 "自证明结构", 在安全性证明中模拟器依旧可提取证据 y 模拟敌手的视角. 因此, 离线批量证明可提升 SM2 适配器签名在预签名阶段的计算效率, 并不影响 SM2-AS2 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 [25, 5], SM2-AS1/2 进行对比. 在线预签名阶段, ECDSA-AS 需要计算预签名公开参数和证明离散对数相等关系, 而根据 SM2 签名和预签名公开参数结构特点, SM2-AS2 可将该部分的计算转移给困难关系选择方离线批量进行, 提升预签名的计算效率, 因此相较于 ECDSA-AS [25, 5], SM2-AS2 在应用中更加高效. 随后, 基于 OpenSSL, 我们实现了 ECDSA-AS [5], SM2-AS1/2, 并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. 具体实现在 Github 上发布: https://github.com/tbb-tobebetter/SM2-adaptor-signature. 实验结果显示 SM2-AS2 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.

实际应用. 我们给出 SM2 适配器签名在区块链中的两种应用: 原子交换协议 [15, 16, 17, 18] 和支付通道网络 [11, 7, 13, 5], 分别实现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. 此外, 我们考虑单参与方与多参与方进行批量公平交换的应用需求, 比如交易所场景, 将两方的原子交换协议扩展为批量原子交换协议, 实现一方 U_0 同时和多方 $U_i, i \in [1, n]$ 进行高效地批量公平交换功能. 根据 ECDSA-AS, SM2-AS1/2 的预签名公开参数结构, ECDSA-AS 的预签名公开参数 $(K = kY)$ 和对应的零知识证明 (随机数 k 作为证据) 只能由预签名方在预签名阶段独自生成, 而 SM2-AS1/2 的预签名公开参数 $(Z = xY + Y)$ 和对应的零知识证明 (签名私钥 x 或困难关系的证据 y 作为证据) 可由预签名方或困难关系选择方生成. 因此, SM2-AS 可根据具体应用情况选择由预签名方或困难关系选择方生成该部分内容, 比 ECDSA-AS 更加灵活. 特别的, SM2-AS2 由困难关系选择方为所有的预签名方离线批量地生成预签名公开参数和公开参数, 避免了各预签名方在预签名阶段自己计算, 相较于 ECDSA-AS 方案, 在批量原子交换协议和中介节点较多的支付通道网络应用中更加高效.

1.2 相关工作

彭聪等 [31] 基于 SM2 签名构造了适配器签名方案. 本文构造的 SM2 适配器签名 SM2-AS1 和彭聪等 [31] 的 SM2 适配器签名方案是并行独立的工作. 不同点在于: (1) 在适配器签名的选择消息攻击下存在不可伪造性的安全性定义中, 我们考虑的敌手模型更强, 该敌手可自适应地选择困难关系和消息访问预签名预言机; (2) 在适配器签名不可伪造性的安全性证明中, 为应对更强能力的敌手, 模拟器需要根据直线提取器提取敌手自适应选择的困难关系 $I_Y = (Y, \pi_Y)$ 的证据 y, 结合 SM2 签名预言机模拟敌手的预签名预言机; (3) 在 SM2-AS1 构造中, 我们根据零知识证明系统是否需要直线证据提取性, 细分了 SM2 适配器签名中使用的两种零知识证明系统, 可保证预签名阶段的零知识证明更加高效. 具体而言, 对于具有 "自证明结构" 的困难关系 $(I_Y = (Y, \pi_Y), y)$, $\pi_Y \leftarrow \mathsf{P}_Y(Y, y)$, 其中 P_Y 是零知识证明系统中的证明算法, 该证明系统需要满足直线证据提取性, 具体构造可参考 [28]. 因为不可伪造性的安全性定义中, 敌手可以自适应的选择消息 m 和实例 I_Y, 所以在安全性证明中, 模拟器需要通过直线证据提取性提取敌手选择的实例的证据 y, 模拟后续敌手对于预签名预言机的询问. 在预签名阶段, 计算预签名公开参数 $Z = xY + Y$ 和对应的零知识证明 $\pi_Z \leftarrow \mathsf{P}_Z(I_Z = (G, X, Y, Z), x)$, 其中 P_Z 是零知识证明系统中的证明算法, 该证明系统不需要满足较强的直线证据提取性. 因为在安全性证明中, 只需要保证模拟器在没有证据 x 的情况下可模拟该证明即可, 不需要提取证据, 具体构造可使用更加高效的零知识证明系统, 比如: 通过 Fiat-Shamir 启发式转换 [29] Chaum-Pedersen $\sum$-协议 [30].

本文的 SM2-AS1 和彭聪等 [31] 的 SM2 适配器签名都是以签名私钥 x 作为证据, 计算预签名公开参数和对应的零知识证明. 因此, 该部分只能由预签名方在获得实例 I_Y 后, 在预签名阶段各自计算预签名公开参数 $Z = (x + 1)Y$ 和对应的零知识证明 π_Z. 随后, 我们进一步提出了离线批量证明技术, 以困难关系 (I_Y, y) 的证据 y 作为证据, 生成预签名公开参数 $Z = y(X + G)$ 和对应零知识证明, 构造了支持离线批量证明的 SM2 适配器签名 SM2-AS2. SM2-AS2 的预签名公开参数和对应的零知识证明可由困难关系

选择方在协议执行前离线批量地为各个预签名参与方生成, 能较大地提升在线预签名操作的计算效率, 在批量原子交换协议和中介节点较多的支付通道网络中的应用更加高效.

2 预备知识

2.1 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 $\{1,\cdots,n\}$. 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 $\mathsf{negl}(\lambda) \geq 0$, 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 $\mathsf{negl}(n) \leq \frac{1}{\mathsf{poly}(n)}$, 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. $y \leftarrow \mathcal{A}(x_1,\cdots,x_t;r)$ 表示 $\mathcal{A}$ 以 $(x_1,\cdots,x_t)$ 和随机数 $r \leftarrow R$ 为输入, 输出 y; 通常我们忽略 r, 记为 $y \leftarrow \mathcal{A}(x_1,\cdots,x_t)$. 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.

2.2 签名

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

- $\mathsf{Gen}(1^\lambda)$: 密钥生成算法输入安全参数 λ, 输出签名密钥对 (vk,sk).
- $\mathsf{Sign}(sk,m)$: 签名算法输入签名私钥 sk 和消息 $m \in \{0,1\}^*$, 输出签名值 σ.
- $\mathsf{Vrfy}(vk,m,\sigma)$: 验证算法输入签名验证公钥 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 $\lambda \in \mathbb{N}$, $(vk,sk) \leftarrow \mathsf{Gen}(1^\lambda)$, $m \in \{0,1\}^*$, 则 $\Pr[\mathsf{Vrfy}(vk,m,\mathsf{Sign}(sk,m)) \to 1] = 1$.

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 $\mathrm{sSigForge}_{\sum,\mathcal{A}}$ 中, 敌手优势 $\mathbf{Adv}_{\mathcal{A},\mathrm{sSigForge}} = \Pr[\mathrm{sSigForge}_{\sum,\mathcal{A}}(\lambda)=1] \leq \mathsf{negl}(\lambda)$, 则称签名方案 $\sum$ 是选择消息攻击下强不可伪造 (strong existential unforgeablity under chosen message attack, SUF-CMA), 其中实验 $\mathrm{sSigForge}_{\sum,\mathcal{A}}$ 定义如下:

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

2.3 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 $x \in [1,n-2]$, 计算签名验证公钥 $X = xG$. 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 $e = H(m)$[3].
2. 选择随机数 $k \in [1,n-1]$, 计算 $r_x = f(kG)$[4].
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.

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

[4]SM2 的变换函数 f 与 ECDSA 类似, 被定义为点 kG 的横坐标, 即 $kG = (r_x, r_y)$, $r_x = f(kG)$ [32, 5].

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 [32, 20, 19].

2.4 困难关系和零知识证明

定义语言 $L_R = \{Y | \exists\ y, \text{s.t.}(Y, y) \in R\}$. R 是困难关系 [5], 如果以下条件成立:

- $\mathsf{GenR}(1^\lambda) \to (Y, y)$: 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 $\mathcal{H}$ 下具有直线证据提取性的非交互零知识的知识证明 (non-interactive zero-knowledge proof of knowledge, NIZKPoK) [5, 28], 需要满足以下条件:

- 完备性: 对于任意困难关系 $(Y, y) \in R$, $\pi_Y \leftarrow \mathsf{P}^{\mathcal{H}}(Y, y)$, 存在可忽略函数 negl, 使得 $\Pr[\mathsf{V}^{\mathcal{H}}(Y, \pi_Y) = 1] \geq 1 - \mathsf{negl}(\lambda)$.
- 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_Y.
- 直线证据提取性: 存在 PPT 提取算法 K, 输入实例 Y 和证明 π_Y, 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 $(Y, y) \in R$.

2.5 适配器签名

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

- $\mathsf{pSign}(vk, sk, m, Y) \to \hat{\sigma}$. 预签名算法输入签名验证公钥 vk、签名私钥 sk、消息 $m \in \{0,1\}^*$, 以及实例 Y, 输出预签名值 $\hat{\sigma}$.
- $\mathsf{pVrfy}(vk, m, Y, \hat{\sigma}) \to 0/1$. 预签名验证算法输入签名验证公钥 vk、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.
- $\mathsf{Adapt}(\hat{\sigma}, y) \to \sigma$. 适配算法输入预签名值 $\hat{\sigma}$ 和证据 y, 输出签名值 σ.
- $\mathsf{Ext}(\sigma, \hat{\sigma}, Y) \to y$. 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.

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

预签名可适配性. 如果对任意的 $\lambda \in \mathbb{N}$, $m \in \{0,1\}^*$, $(Y, y) \in R$, $(vk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$, 以及任意可验证成功的预签名 $\hat{\sigma}$, $\mathsf{pVrfy}(vk, m, Y, \hat{\sigma}) \to 1$, 都有 $\Pr[\mathsf{Vrfy}(vk, m, \mathsf{Adapt}(\hat{\sigma}, y)) \to 1] = 1$, 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 unforgeablity 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, aWitExt). 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 $\mathbf{Adv}_{\mathcal{A},\mathrm{aSigForge}} = \Pr[\mathrm{aSigForge}_{\mathcal{A},\Pi_{R,\sum}}(\lambda)=1] \leq \mathsf{negl}(\lambda)$ 成立, 则称适配器签名满足 aEUF–CMA, 其中 $\mathrm{aSigForge}_{\mathcal{A},\Pi_{R,\sum}}$ 定义如下:

$\mathrm{aSigForge}_{\mathcal{A},\Pi_{R,\sum}}(\lambda)$	$\mathcal{O}_{\mathrm{S}}(m)$
$\mathcal{Q}=\emptyset$	$\sigma \leftarrow \mathsf{Sign}(sk,m)$
$(vk,sk) \leftarrow \mathsf{Gen}(1^\lambda)$	$\mathcal{Q}=\mathcal{Q}\cup\{m\}$
$m \leftarrow \mathcal{A}^{\mathcal{O}_{\mathrm{S}}(\cdot),\mathcal{O}_{\mathrm{pS}}(\cdot)}(vk)$	return σ
$(Y,y) \leftarrow \mathsf{GenR}(1^\lambda)$	
$\hat{\sigma} \leftarrow \mathsf{pSign}((vk,sk),Y,m)$	$\mathcal{O}_{\mathrm{pS}}(m,Y)$
$\sigma \leftarrow \mathcal{A}^{\mathcal{O}_{\mathrm{S}}(\cdot),\mathcal{O}_{\mathrm{pS}}(\cdot)}(\hat{\sigma},Y)$	$\hat{\sigma} \leftarrow \mathsf{pSign}((vk,sk),m,Y)$
return $(m \notin \mathcal{Q} \wedge \mathsf{Vrfy}(vk,m,\sigma))$	$\mathcal{Q}=\mathcal{Q}\cup\{m\}$
	return $\hat{\sigma}$

除了提供额外的预签名预言机 $\mathcal{O}_{\mathrm{pS}}(\cdot)$, 适配器签名的 aEUF–CMA 定义与签名的不可伪造性安全定义类似. 预签名预言机输入消息值 m 和实例 Y 输出一个预签名值, 其中 (m,Y) 可由敌手自适应选择. 预签名预言机是非常重要的, 因为在具体应用中适配器签名要求敌手在不知道实例 Y 的证据 y 的情况下, 即使知道预签名值 $\hat{\sigma}$ 也难以伪造消息的真实签名值 σ. aEUF–CMA 安全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 $\mathrm{aWitExt}_{\mathcal{A},\Pi_{R,\sum}}$ 中敌手优势 $\mathbf{Adv}_{\mathcal{A},\mathrm{aWitExt}} = \Pr[\mathrm{aWitExt}_{\mathcal{A},\Pi_{R,\sum}}(\lambda)=1] \leq \mathsf{negl}(\lambda)$, 则称适配器签名满足证据可提取性. $\mathrm{aWitExt}_{\mathcal{A},\Pi_{R,\sum}}$ 定义如下:

$\mathrm{aWitExt}_{\mathcal{A},\Pi_{R,\sum}}(\lambda)$	$\mathcal{O}_{\mathrm{S}}(m)$
$\mathcal{Q}=\emptyset$	$\sigma \leftarrow \mathsf{Sign}(sk,m)$
$(vk,sk) \leftarrow \mathsf{Gen}(1^\lambda)$	$\mathcal{Q}=\mathcal{Q}\cup\{m\}$
$(m,Y) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathrm{S}}(\cdot),\mathcal{O}_{\mathrm{pS}}(\cdot)}(vk)$	return σ
$\hat{\sigma} \leftarrow \mathsf{pSign}((vk,sk),Y,m)$	
$\sigma \leftarrow \mathcal{A}^{\mathcal{O}_{\mathrm{S}}(\cdot),\mathcal{O}_{\mathrm{pS}}(\cdot)}(\hat{\sigma})$	$\mathcal{O}_{\mathrm{pS}}(m,Y)$
$y' \leftarrow \mathsf{Ext}(\sigma,\hat{\sigma},Y)$	$\hat{\sigma} \leftarrow \mathsf{pSign}((vk,sk),m,Y)$
return $(m \notin \mathcal{Q} \wedge (Y,y') \notin R$	$\mathcal{Q}=\mathcal{Q}\cup\{m\}$
$\wedge \mathsf{Vrfy}(vk,m,\sigma)$	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, 敌手选择挑战实例 Y, 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. 但是这个优势不足以让敌手赢得实验 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 $\sum_{\mathrm{AS}}$ 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 $\sum_{\mathrm{AS}}$ 是安全的.

3 SM2 适配器签名

3.1 方案设计

本节中, 我们构造 SM2 适配器签名方案 SM2-AS1, 并在随机预言机模型下给出安全性证明. 令困难关系 $R = \{(I_Y = (Y, \pi_Y), y) \mid Y = yG \wedge \mathsf{V}_Y(Y, \pi_Y) \to 1\}$, 其中 $\pi_Y \leftarrow \mathsf{P}_Y(Y, y)$, P_Y 和 V_Y 分别表示具有直线证据提取性的非交互零知识的知识证明 (NIZKPoK$_Y$) [28] 中的证明和验证算法. 令困难关系 $R_Z = \{(I_Z = (G, X, Y, Z), x) \mid X = xG \wedge Z - Y = xY\}$, 其中 P_Z 和 V_Z 分别表示非交互零知识证明 (NIZK$_Z$) [30] 中的证明和验证算法. SM2 适配器签名算法构造如下, 具体见图 1:

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

$\mathsf{pSign}((X, x), m, I_Y)$	$\mathsf{pVrfy}(X, m, I_Y, \hat{\sigma})$	$\mathsf{Adapt}(\hat{\sigma}, y)$
$vk = X, sk = x$	$vk = X, I_Y = (Y, \pi_Y)$	$\hat{\sigma} = (r, \hat{s}, Z, \pi_Z)$
$I_Y = (Y, \pi_Y)$	$\hat{\sigma} = (r, \hat{s}, Z, \pi_Z)$	$s = \hat{s} + y \bmod n$
$Z = (x + 1)Y$	$b_Z \leftarrow \mathsf{V}_Z(I_Z, \pi_Z)$	return $\sigma = (r, s)$
$\mathsf{P}_Z(I_Z, x) \to \pi_Z$	$e = H(m)$	
$e = H(m)$	$\hat{K} = (\hat{s} + r)X + \hat{s}G + Z$	$\mathsf{Ext}(\sigma, \hat{\sigma}, I_Y)$
$k \leftarrow \mathbb{Z}_n$	$r' = f(\hat{K}) + e \bmod n$	$\sigma = (r, s)$
$\hat{K} = kG + Z, r_x = f(\hat{K})$	$b_r = (r' = r)$	$\hat{\sigma} = (r, \hat{s}, Z, \pi_Z)$
$r = e + r_x$	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 SM2 适配器签名 (SM2-AS1)
Figure 1 SM2-based adaptor signature scheme (SM2-AS1)

3.2 安全性证明

定理 1 如果 SM2 签名 $\sum_{\mathrm{SM2}}$ 满足 SUF–CMA, R 是一个困难关系, NIZKPoK$_Y$ 是安全的具有直线证据提取性的非交互零知识的知识证明, NIZK$_Z$ 是安全的非交互零知识证明, 则上述 SM2 适配器签名方案 $\Pi_{R,\sum_{\mathrm{SM2}}}$ 在随机预言机模型下是安全的.

引理 1 (预签名可适配性.) SM2 适配器签名 $\Pi_{R,\sum_{\mathrm{SM2}}}$ 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 $(I_Y, y) \in R$, SM2 签名验证公钥 $X \in \mathbb{G}$ 和有效的预签名值 $\hat{\sigma} = (r, \hat{s}, Z, \pi_Z)$. 预签名值能通过预签名值验证算法 $\mathsf{pVrfy}(X, m, I_Y, \hat{\sigma}) \to 1$. 即 $\hat{K} = (\hat{s} + r)X + \hat{s}G + Z = kG + Z$, $r' = r'_x + e = f(\hat{K}) + e = f(kG + Z) + e = r \bmod n$.

根据适配算法的定义, $\mathsf{Adapt}(\hat{\sigma}, y) \to \sigma$, 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 $r' = r'_x + e = f(K') + e = f(kG + Z) + e = r \bmod n$. 即预签名值能被适配成 SM2 签名值, 并能通过 SM2 的签名验证算法 $\mathsf{Vrfy}(vk, m, \sigma) \to 1$. □

引理 2 (预签名的正确性.) SM2 适配器签名 $\Pi_{R,\sum_{\mathrm{SM2}}}$ 满足预签名的正确性.

证明: 对于任意的签名私钥和证据 $x, y \in \mathbb{Z}_n$, 消息 $m \in \{0,1\}^*$, 定义签名验证公钥和实例 $X = xG, Y = yG, \pi_Y \leftarrow \mathsf{P}_Y(Y, y), I_Y = (Y, \pi_Y)$. 根据预签名算法的定义 $\mathsf{pSign}((X,x), m, I_Y) \rightarrow (r, \hat{s}, Z, \pi_Z)$, 对于随机数 $k \leftarrow \mathbb{Z}_n$, $\hat{s} = (1+x)^{-1}(k - rx) \bmod n$, $K = (\hat{s} + r)X + \hat{s}G = kG$.

根据 NIZK_Z 的完备性, $\mathsf{V}_Z(I_Z, \pi_Z) \rightarrow 1$, 则有 $Z = (x+1)Y = (x+1)yG$, $\hat{K} = K + Z = (k + y(x+1))G, r'_x = f(\hat{K}), r' = r'_x + e = r$, 即可通过预签名验证算法 $\mathsf{pVrfy}(X, m, I_Y, \hat{\sigma}) \rightarrow 1$. 由于 $\mathsf{Adapt}(\hat{\sigma}, y) \rightarrow \sigma$, 根据适配算法 Adapt 的定义 $s = \hat{s} + y$, 可通过签名验证算法 $\mathsf{Vrfy}(vk, m, \sigma) \rightarrow 1$. 同时可提取正确的证据 $y = s - \hat{s}$, 即 $\mathsf{Ext}(\sigma, \hat{\sigma}, I_Y) = \hat{s} + y - \hat{s} = y$. □

引理 3 (*aEUF–CMA* 安全.) 如果 SM2 签名算法 $\sum_{\mathrm{SM2}}$ 满足 SUF–CMA, R 是一个困难关系, $\mathrm{NIZKPoK}_Y$ 是安全的具有直线证据提取性的非交互零知识的知识证明, NIZK_Z 是安全的非交互零知识证明, 则 SM2 适配器签名 $\Pi_{R,\sum_{\mathrm{SM2}}}$ 满足 aEUF–CMA 安全.

证明: 我们在 ROM 模型下将 SM2 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 $\mathcal{H}_{\mathrm{SM2}}$, 并且模拟 $\mathcal{A}$ 的随机预言机 ($\mathcal{H}$), 签名预言机 ($\mathcal{O}_{\mathrm{S}}$) 和预签名预言机 ($\mathcal{O}_{\mathrm{pS}}$).

模拟器 $\mathcal{S}$ 可以通过 $\mathcal{O}_{\text{SM2-Sign}}$ 和 $\mathcal{H}_{\mathrm{SM2}}$ 直接模拟 $\mathcal{A}$ 的 $\mathcal{O}_{\mathrm{S}}$ 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 $\mathcal{O}_{\mathrm{pS}}$ 较为困难. 具体困难包括: 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. 上述困难可解决如下: 当 $\mathcal{A}$ 询问 $\mathcal{O}_{\mathrm{pS}}$ 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 (r, s), 并通过 $\mathrm{NIZKPoK}_Y$ 的直线证据提取器提取实例 (Y, π_Y) 的证据 y, 并根据适配器算法的定义, 基于证据 y 和真实签名值 (r, s) 计算预签名值 $(r, \hat{s} = s - y)$, 最后, $\mathcal{S}$ 利用 NIZK_Z 的零知识性, 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 $\mathcal{O}_{\mathrm{pS}}$.

通过混合论证技术, 我们首先描述 5 个游戏 $\mathrm{Game}_0, \cdots, \mathrm{Game}_4$, 其中 Game_0 是真实的 aSigForge 实验, Game_i 和 Game_{i+1}, $i = 0, \cdots, 3$ 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: 该游戏和真实的适配器签名 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 $\sigma^* = \mathsf{Adapt}(\hat{\sigma}, y)$, 则输出 $\perp$.

Game_2: 该游戏和 Game_1 的区别是在敌手询问 $\mathcal{O}_{\mathrm{pS}}$ 时, Game_2 使用 $\mathrm{NIZKPoK}_Y$ 的直线证据提取器提取证据 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 $\mathcal{O}_{\mathrm{pS}}$ 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 $Z = y(X + G)$, 模拟 NIZK_Z 的零知识证明为 $\pi_S \leftarrow \mathsf{S}((G, X, Y, Z), 1)$, 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 NIZK_Z 的零知识证明为 $\pi^*_S \leftarrow \mathsf{S}((G, X, Y^*, Z^*), 1)$, 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 SM2 签名的 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 $\mathcal{H}_{\mathrm{SM2}}$ 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 $\mathrm{NIZKPoK}_Y$ 的直线证据提取器提取证据 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 NIZK_Z 的零知识证明为 π_S, 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 $\mathsf{GenR}(1^\lambda) \to (I_Y, y)$, 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 NIZK_Z 的零知识证明 π_S, 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^*).

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

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

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 I_Y^*, 使用 Game_1 中的方式模拟 $\mathcal{A}$ 的 $\mathcal{H}$, $\mathcal{O}_{\text{S}}$ 和 $\mathcal{O}_{\text{pS}}$. 当收到 $\mathcal{A}$ 的挑战消息 m 时, 以 I_Y^* 作为实例计算预签名值 $\hat{\sigma}$ 并发送给敌手. 随后获得敌手输出的完整签名值 σ.

假设事件 Bad_1 发生, 即 $\mathsf{Adapt}(\hat{\sigma}, y) = \sigma$, 因此 $\mathcal{S}$ 可以提取证据 $y^* \leftarrow \mathsf{Ext}(\sigma, \hat{\sigma}, I_Y^*)$, 可获得证据 y^* 使得 $(I_Y^*, y^*) \in R$ 成立. 因此, $\mathcal{S}$ 打破困难关系的概率和 Bad_1 发生的概率相同, 即 $\Pr[\text{Bad}_1] \leq \mathsf{negl}_1(\lambda)$. 除了 Bad_1 发生, Game_1 和 Game_0 是相同的, 即 $|\Pr[\text{Game}_0 = 1] - \Pr[\text{Game}_1 = 1]| \leq \mathsf{negl}_1(\lambda)$. □

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

证明: 根据 NIZKPoK_Y 的直线证据提取性, 基于实例 Y 及证明 π_Y 可提取证据 y, 使得 $\mathsf{V}_Y(Y, \pi_Y) \to 1$ 成立, 且 $(I_Y, y) \in R$ 成立. 因此, Game_2 和 Game_1 是不可区分的, 即 $|\Pr[\text{Game}_2 = 1] - \Pr[\text{Game}_1 = 1]| \leq \mathsf{negl}_2(\lambda)$.

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

在 Game_4 中, $\mathcal{S}$ 接收到 $\mathcal{A}$ 的挑战消息 m^*, 首先生成 $(I_Y, y) \leftarrow \mathsf{GenR}(1^\lambda)$, 计算完整签名值 $\sigma = (r, s)$, 和预签名值 $\hat{s} = s - y$, 并模拟证明 $\pi_S \leftarrow \mathsf{S}((G, X, Y, Z), 1)$, 输出预签名值 $\hat{\sigma} = (r, \hat{s}, Z, \pi_S)$. 由于 NIZK_Z 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 $|\Pr[\text{Game}_4 = 1] - \Pr[\text{Game}_3 = 1]| \leq \mathsf{negl}_4(\lambda)$. □

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 $\mathcal{O}_{\mathrm{S}}$ 或者 $\mathcal{O}_{\mathrm{pS}}$ 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. □

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

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

因此, 上述 SM2 适配器签名满足 aEUF–CMA 安全. □

引理 4 (证据可提取性.) 假设 SM2 签名满足 SUF–CMA 安全, R 是一个困难关系, $\mathrm{NIZKPoK}_Y$ 是安全的具有直线证据提取性的非交互零知识的知识证明, NIZK_Z 是安全的非交互零知识证明, 则 SM2 适配器签名 $\Pi_{R,\sum_{\mathrm{SM2}}}$ 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 SM2 适配器签名的 aWitExt 实验, 则可构造 PPT 的模拟器赢得 SM2 签名的 sSigForge 实验.

不同于实验 aSigForge, 在实验 aWitExt 中, 挑战阶段的困难关系实例 I_Y 由敌手 $\mathcal{A}$ 生成. 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 $\mathrm{NIZKPoK}_Y$ 的直线证据提取器提取 I_Y 中的证据 y, $\mathcal{S}$ 同样可模拟敌手的预签名值预言机 $\mathcal{O}_{\mathrm{pS}}$.

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

Game_0: 该游戏和真实的 SM2 适配器签名的 aWitExt 实验相同.

Game_1: 该游戏和 Game_0 的区别是在模拟 $\mathcal{O}_{\mathrm{pS}}$ 时, 通过 $\mathrm{NIZKPoK}_Y$ 的直线证据提取器 K 提取实例 I_Y 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 $\mathcal{O}_{\mathrm{pS}}$ 时, 首先生成签名值 $\sigma = (r, s)$, 使用证据 y 计算 $\hat{s} = s - y$, 然后计算 $Z = y(X + G)$ 并基于 NIZK_Z 的零知识性模拟证明 π_S, 输出预签名值 $\hat{\sigma} = (r, \hat{s}, Z, \pi_S)$.

Game_3: 该游戏和 Game_2 的区别是在挑战阶段使用 $\mathrm{NIZKPoK}_Y$ 的直线证据提取器提取证据 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) 后, 首先生成签名值 $\sigma = (r, s)$, 使用证据 y 计算 $\hat{s} = s - y$, 然后计算 $Z = y(X + G)$ 并基于 NIZK_Z 的零知识性模拟证明 π_S, 最后输出预签名值 $\hat{\sigma} = (r, \hat{s}, Z, \pi_S)$.

可构造模拟器 $\mathcal{S}$ 完美模拟 Game_4, 并利用 $\mathcal{A}$ 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 $\mathcal{H}_{\mathrm{SM2}}$ 回复, 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 $\mathrm{NIZKPoK}_Y$ 的直线证据提取器提取证据 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). 然后计算 $\hat{s} = s - y, Z = y(G + X)$, 并基于 NIZK_Z 的零知识性模拟证明 π_S. 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 NIZKPoK$_Y$ 的直线证据提取器提取证据 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 $\hat{s}=s-y, Z=y(G+X)$, 并基于 NIZK$_Z$ 的零知识性模拟证明 π_S, 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^*).

$\mathcal{S}^{\mathcal{O}_{\text{SM2-Sign}},\mathcal{H}_{\text{SM2}}}(vk)$	$\mathcal{H}(x)$
$\mathcal{Q}=\emptyset$	if $H[x]=\perp$
$H=[\perp]$	$H[x]\leftarrow\mathcal{H}_{\text{SM2}}(x)$
$(m^*,I_Y)\leftarrow\mathcal{A}^{\mathcal{O}_{\text{S}}(\cdot),\mathcal{O}_{\text{pS}}(\cdot)}(vk)$	return $H[x]$
parse I_Y as (Y,π_Y)	
$y\leftarrow\mathsf{K}(Y,\pi_Y,H)$	$\mathcal{O}_{\text{pS}}(m,I_Y)$
if $((Y,\pi_Y),y)\notin R$	parse I_Y as (Y,π_Y)
abort	$y\leftarrow\mathsf{K}(Y,\pi_Y,H)$
$\sigma\leftarrow\mathcal{O}_{\text{SM2-Sign}}(m^*)$	if $((Y,\pi_Y),y)\notin R$
parse σ as (r,s)	abort
$\hat{s}=s-y$	$\sigma\leftarrow\mathcal{O}_{\text{SM2-Sign}}(m)$
$Z=y(X+G)$	parse σ as (r,s)
$\pi_S\leftarrow\mathsf{S}((G,X,Y,Z),1)$	$\hat{s}=s-y$
$\hat{\sigma}=(r,s,Z,\pi_S)$	$Z=y(X+G)$
$\sigma^*\leftarrow\mathcal{A}^{\mathcal{O}_{\text{S}}(\cdot),\mathcal{O}_{\text{pS}}(\cdot)}(\hat{\sigma},I_Y)$	$\pi_S\leftarrow\mathsf{S}((G,X,Y,Z),1)$
return (m^*,σ^*)	$\mathcal{Q}=\mathcal{Q}\cup\{m\}$
$\mathcal{O}_{\text{S}}(m)$	return $(r,\hat{s},Z,\pi_S)$
$\sigma\leftarrow\mathcal{O}_{\text{SM2-Sign}}(m)$	
$\mathcal{Q}=\mathcal{Q}\cup\{m\}$	
return 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 NIZKPoK$_Y$ 的直线证据提取性, 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 $\mathsf{V}_Y(Y,\pi_Y)=1$, 且 $((Y,\pi_Y),y)\in R$, 因此, Game_1 和 Game_0 是不可区分的, 即 $|\Pr[\text{Game}_0=1]-\Pr[\text{Game}_1=1]|\leq\mathsf{negl}_1(\lambda)$.

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 $|\Pr[\text{Game}_2=1]-\Pr[\text{Game}_3=1]|\leq\mathsf{negl}_3(\lambda)$.

根据 NIZK$_Z$ 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 $\pi_S\leftarrow\mathsf{S}((G,X,Y,Z),1)$, 且 π_S 和真实产生的证明 $\pi_Z\leftarrow\mathsf{P}((G,X,Z,Y),x)$ 不可区分. 因此, Game_2 和 Game_1 是不可区分的, 即, $|\Pr[\text{Game}_1=1]-\Pr[\text{Game}_2=1]|\leq\mathsf{negl}_2(\lambda)$.

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 $|\Pr[\text{Game}_3=1]-\Pr[\text{Game}_4=1]|\leq\mathsf{negl}_4(\lambda)$. □

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 $\mathcal{O}_{\text{S}}$ 或者 $\mathcal{O}_{\text{pS}}$ 关于挑战消息 m^* 的签名值或预签名值. 关于 m^* 对于 $\mathcal{O}_{\text{SM2-Sign}}$ 的询问发生在挑战阶段. 如果在挑战阶段敌手输出的 SM2 签名伪造 σ^* 和模拟器询问 $\mathcal{O}_{\text{SM2-Sign}}$ 获得的 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, $\mathcal{O}_{\text{SM2-Sign}}$ 从没有输出过 σ^*, 即 (m^*,σ^*) 是实验 sSigForge 中的有效伪造. □

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

验aWitExt中的优势为:

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

因此, 上述 SM2 适配器签名满足证据可提取性.

□

4 支持离线批量证明的 SM2 适配器签名

本节中, 我们提出离线批量证明技术, 在 SM2-AS1 的基础上构造更加高效的支持离线批量证明的 SM2 适配器签名 SM2-AS2. 根据 SM2 适配器签名的预签名公开参数 $Z = (x+1)Y = y(X+G)$, 可分别以 x 和 y 为证据计算和证明 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选择方生成. 由于证据 y 由困难关系选择方生成, 签名公钥 X 和基点 G 都是公开信息. 因此, 预签名公开参数和对应的零知识证明可由困难关系选择方离线批量生成.

令困难关系 $R = \{(I_Y = (Y, \pi_Y), y) \,|Y = yG \wedge \mathsf{V}_Y(I_Y) \to 1\}$, 其中 $\pi_Y \leftarrow \mathsf{P}_Y(Y, y)$, P_Y 和 V_Y 分别表示具有直线证据提取性的非交互零知识的知识证明 (NIZKPoK$_Y$) [28] 中的证明和验证算法. 令困难关系 $R_Z = \{(I_Z = (G, X, Y, Z, \pi_Z), x)|X = xG \wedge Z - Y = xY \wedge \mathsf{V}_Z(I_Z) \to 1\}$, 其中 $\pi_Z \leftarrow \mathsf{P}_Z(I_Z, y)$, P_Z 和 V_Z 分别表示非交互零知识证明 (NIZK$_Z$) [30] 中的证明和验证算法. 令 $I = (I_Y, I_Z)$. 支持离线批量证明的 SM2 适配器签名 SM2-AS2 构造如下, 具体见图 2:

- $\mathsf{pSign}(x, m, I) \to \hat{\sigma}$. 预签名算法输入签名私钥 x, 消息值 m 和实例 I=(I_Y,I_Z), 计算 $e = H(m)$, 选择随机数 $k \leftarrow \mathbb{Z}_n$, 计算 $\hat{K} = kG + Z$, $r_x = f(\hat{K})$, $r = e + r_x \bmod n$, $\hat{s} = (1+x)^{-1}(k - rx) \bmod n$, 最后输出预签名值 $\hat{\sigma} = (r, \hat{s})$.
- $\mathsf{pVrfy}(X, m, I, \hat{\sigma}) \to 0/1$. 预签名验证算法输入验证公钥 X, 消息 m, 实例 I=(I_Y,I_Z), 以及预签名值 $\hat{\sigma} = (r, \hat{s})$, 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.
- $\mathsf{Adapt}(\hat{\sigma}, y) \to \sigma$. 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 $\sigma = (r, s)$.
- $\mathsf{Ext}(\sigma, \hat{\sigma}, I) \to y$. 证据提取算法输入签名值 σ, 预签名值 $\hat{\sigma}$ 和实例 I=(I_Y,I_Z), 计算 $y = s - \hat{s} \bmod n$. 如果 $(I_Y, y) \in R$, 输出 y, 否则输出 $\perp$.

$\mathsf{pSign}(x, m, I) \to \hat{\sigma}$	$\mathsf{pVrfy}(X, m, I, \hat{\sigma}) \to 0/1$	$\mathsf{Adapt}(\hat{\sigma}, y) \to \sigma$
$e = H(m)$	$e = H(m)$	$s = \hat{s} + y \bmod n$
$k \leftarrow \mathbb{Z}_n$	$\hat{K} = (\hat{s} + r)X + \hat{s}G + Z$	**reture** $\sigma = (r, s)$
$r_x = f(kG + Z)$	$r' = f(\hat{K}) + e \bmod n$	
$r = e + r_x \bmod n$	If $r' = r$, output 1,	$\mathsf{Ext}(\sigma, \hat{\sigma}, I) \to y$
$\hat{s} = (1+x)^{-1}(k - rx) \bmod n$	else, output 0.	$y = s - \hat{s} \bmod n$
reture $\hat{\sigma} = (r, \hat{s})$	**reture** 0/1	If $(I_Y, y) \in R$, output y, else outputs $\perp$
		reture $y/\perp$

图 2 支持离线批量证明的 SM2 适配器签名 (SM2-AS2)
Figure 2 SM2-based adaptor signature scheme with offline batch proof (SM2-AS2)

根据是否对消息进行预签名, 可将适配器签名分为两个不同的阶段, 即离线阶段和在线阶段. 对于 SM2-AS2, 在离线阶段, 困难关系选择方可以生成困难关系 (I_Y, y), 并根据证据 y 以及各个预签名方 U_i 的签名验证公钥 X_i 批量生成预签名公开参数 $Z_i = y(X_i + G)$ 和对应的零知识证明 π_{Z_i}, 发送给各个预签名方. 预签名方可验证 π_{Z_i} 的正确性, 并存储预签名公开参数 Z_i 用于后续消息的预签名. 此外, 根据具体应用情况, 可根据预签名方生成多个预签名值的需求, 困难关系选择方可批量生成对应数量的预签名公开参数. 在线阶段, 各预签名方使用存储的预签名公开参数 Z_i 和签名私钥对消息 m 进行预签名, 不需要额外计算 Z_i 和零知识证明.

定理 2 如果 SM2 签名满足 SUF−CMA, R 是一个困难关系, NIZKPoK$_Y$ 是安全的具有直线证据提取性的非交互零知识的知识证明, NIZK$_Z$ 是安全的非交互零知识证明, 则上述 SM2 适配器签名方案 SM2-AS2 在随机预言机模型下是安全的.

证明: SM2-AS2 和 SM2-AS1 的区别只在于生成预签名公开参数的方式不同: $Z = (x+1)Y = y(X+G)$, 并不影响适配器签名的正确性和安全性. 因此, 根据引理 1 和 2, SM2-AS2 也满足预签名可适配性和预签名正确性. 根据引理 3 和引理 4, SM2-AS2 满足 aEUF–CMA 安全和证据可提取性. □

对比 ECDSA-AS 和 SM2-AS1. 适配器签名方案 SM2-AS1/2 和 ECDSA-AS 是 SM2/ECDSA 签名算法对于离散对数困难关系 (Y, y) 的扩展. 为了保证可证明安全, 三个方案都要求困难关系满足 "自证明结构", 即在实例中增加额外的零知识证明, 将 (Y, y) 扩展为 $(I_Y = (Y, \pi_Y), y)$, $\pi_Y \leftarrow \mathsf{P}_Z(Y, y)$, 其中 NIZKPoK 需要满足直线证据提取性. 该结构能使得在安全性证明中, 模拟器可提取敌手自适应选择的困难关系的证据, 模拟敌手的预签名预言机.

在预签名过程中, 三个方案都需要额外的预签名公开参数, 以及对应的零知识证明保证预签名公开参数的正确性. 不同的是, ECDSA-AS 以预签名随机数 k 为证据, 计算预签名公开参数 $K = kY$, 并证明 $\hat{K} = kG$ 和 K 满足相同的离散对数关系; SM2-AS1 以签名私钥 x 为证据, 计算预签名公开参数 $Z = (x+1)Y$, 并证明 SM2 签名验证公钥 X 和 $Z - Y$ 满足相同的离散对数关系; SM2-AS2 以困难关系的证据 y 为证据, 计算预签名公开参数 $Z = y(X+G)$, 并证明实例 Y 和 $Z - Y$ 满足相同的离散对数关系.

根据上述分析, SM2-AS1 和 ECDSA-AS 的预签名公开参数和对应的零知识证明只能由预签名方 (签名私钥 x 和拥有随机数 k) 各自生成, 但是 SM2-AS2 的预签名公开参数和对应的零知识证明可以由困难关系选择方 (拥有证据 y) 批量生成. 通过离线/在线阶段划分, SM2-AS2 可将本应由各预签名方独自生成的预签名公开参数和对应的零知识证明, 统一转移到由困难关系选择方离线批量生成, 可提高预签名操作的计算效率, 适用于批量原子交换和中介节点较多的支付通道网络.

5 方案效率和实验结果

5.1 理论分析

本节分别在通信成本和计算效率两个方面对 ECDSA-AS, SM2-AS1 和 SM2-AS2 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 [25] 提出了首个 ECDSA-AS 构造并未给出安全性证明. 随后 Aumayr 等 [5] 在 [25] 的基础上, 提出了自证明结构 $\{(I_Y = (Y, \pi_Y), y) | Y = yG \wedge \mathsf{V}(Y, \pi_Y) \rightarrow 1\}$, 并给出了可证明安全的 ECDSA-AS. 但是该方案在预签名过程中, 基于预签名随机数生成预签名公开参数及对应的零知识证明. 一方面, 该部分只能由预签名方生成, 需要 4 次点乘操作. 另一方面, 每次重新进行预签名时, 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, ECDSA-AS 的预签名过程较为复杂, 应用不够灵活.

我们沿用自证明结构 [5] 设计了 SM2 适配器签名 SM2-AS1. 该方案在预签名过程中以签名私钥为证据, 计算预签名公开参数和对应的零知识证明, 每次预签名计算依旧需要 4 次点乘操作. 随后, 我们基于 SM2 签名算法的结构特性, 构造了支持离线批量证明的 SM2 适配器签名 SM2-AS2. SM2-AS2 以 y 为证据, 计算预签名公开参数和对应的零知识证明. 该部分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, 而且该部分可由困难关系选择方离线批量生成. 因此, SM2-AS2 避免了在线签名阶段计算预签名公开参数和对应的零知识证明, 在线预签名计算只需要 1 次点乘操作, 较大地提升了在线预签名的效率. 此外, 预签名公开参数和对应的零知识证明已在预签名前的离线阶段完成传输和验证, 所以预签名值不需要额外传输该部分内容, 包括一个椭圆曲线上的点和两个域上的元素. 具体如表 1所示.

5.2 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-AS1 和 SM2-AS2, 具体实现在 Github 上发布: https://github.com/tbb-tobebetter/SM2-adaptor-signature. 执行环境为: Intel Core i5 CPU 2.3 GHz, 8GB RAM, macOS High Sierra 10.13.3 system.

表 1 通信成本与在线效率对比
Table 1 Communication cost and online 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 [25]	$\|\mathbb{G}\|$	$\|\mathbb{Z}_n\|$	$4\|\mathbb{Z}_n\|+\|\mathbb{G}\|$	4Exp	6Exp	?
ECDSA-AS [5]	$\|\mathbb{G}\|$	$\|\mathbb{Z}_n\|$	$4\|\mathbb{Z}_n\|+\|\mathbb{G}\|$	4Exp	6Exp	√
SM2-AS1	$\|\mathbb{G}\|$	$\|\mathbb{Z}_n\|$	$4\|\mathbb{Z}_n\|+\|\mathbb{G}\|$	4Exp	6Exp	√
SM2-AS2	$\|\mathbb{G}\|$	$\|\mathbb{Z}_n\|$	$2\|\mathbb{Z}_n\|$	Exp	6Exp	√

‡ 其中, $|\mathbb{G}|$ 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.

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-AS1 和 SM2-AS2 的在线预签名操作单次平均耗时分别为 166.54μs, 176.27μs, 37.14μs; 预签名验证操作单次平均耗时分别为 252.86μs, 238.46μs, 234.80μs. 实验结果符合上述理论分析, SM2-AS2 的在线预签名耗时仅为 ECDSA-AS 的 1/4.

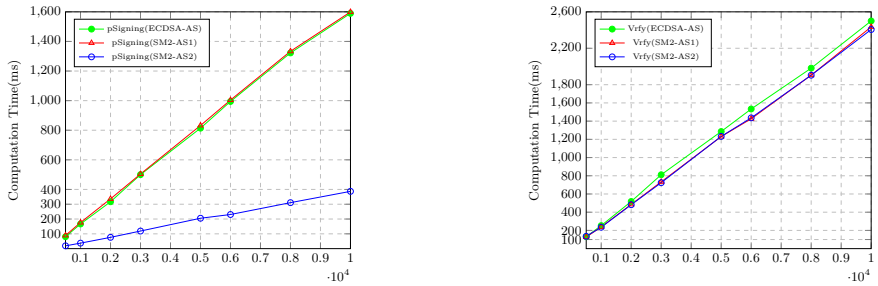

图 3 (a) 预签名算法和 (b) 验证算法效率对比
Figure 3 Efficiency comparison of (a) pre-signing and (b) verification algorithm

6 应用

在本节中, 我们根据现有适配器签名应用 [16, 13, 27, 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. 此外, 在两方原子交换协议的基础上, 我们考虑多方场景, 给出批量原子交换协议, 并基于 SM2 适配器签名给出具体的构造.

6.1 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_0 和 U_1 想要公平地交换不同的密码货币 c_0 和 c_1, 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. 基于适配器签名设计原子交换协议框架 [15, 16, 27], 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:

- 协议建立阶段: U_0 运行 $\mathsf{GenR}(1^\lambda)\to(Y,y)$ 生成困难关系 (Y,y), 计算零知识证明 $\mathsf{P}(Y,y)\to\pi_Y$, 令 $I=(Y,\pi_Y)$. 生成关于传输货币 c_0 给 U_1 的交易 tx_0, 并生成预签名 $\hat{\sigma_0}\leftarrow\mathsf{pSign}((X_0,x_0),tx_0,I)$; 将 $I,\hat{\sigma_0},tx_0$ 发送给 U_1; U_1 可验证实例 I 和预签名值 $\hat{\sigma_0}$, 如果 $\mathsf{V}(I)\to 0$ 或者 $\mathsf{pVrfy}(X_0,tx_0,I,\hat{\sigma_0})\to 0$, 拒绝交易; 否则生成关于传输货币 c_1 给 U_0 的交易 tx_1, 并生成预签名 $\hat{\sigma_1}\leftarrow\mathsf{pSign}((X_1,x_1),tx_1,I)$; 将 $\hat{\sigma_1},tx_1$ 发送给 U_0;
- 交换阶段: U_0 验证预签名值, 如果 $\mathsf{pVrfy}(X_1,tx_1,I,\hat{\sigma_1})\to 0$, 则拒绝交易; 否则根据证据 y, 运行适配算法 $\sigma_1\leftarrow\mathsf{Adapt}(\hat{\sigma_1},y)$ 将 U_1 的预签名适配成 SM2 签名, 即获得 U_1 传输货币 c_1 给 U_0 的 SM2

签名值 σ_1, 在链上公布 σ_1 可获得 c_1; U_1 获得签名值 σ_1, 运行提取算法获得证据 $y \leftarrow \mathsf{Ext}(\sigma_1, \hat{\sigma_1}, I)$, 并运行适配算法 $\sigma_0 \leftarrow \mathsf{Adapt}(\hat{\sigma_0}, y)$, 获得 U_0 传输货币 c_0 给 U_1 的 SM2 签名值 σ_0, 在链上公布 σ_0 获得 c_0, 完成公平交换.

上述原子交换协议只能实现两方的公平交换, 但是在面向交易用户较大的场景, 比如交易所等, 需要批量向多数用户同时进行公平交换, 或者单一参与方同时向多个账户 (地址) 进行公平交换的场景, 两方的原子交换协议效率较低. 在上述两方原子交换协议的基础上, 我们可考虑发起方 U_0 向多个交换方 $U_i, i \in [1, n]$ 进行跨链货币公平交换的扩展协议——批量原子交换协议.

批量原子交换. 在协议建立阶段, U_0 生成困难关系 $(I = (Y, \pi_Y), y)$ 以及向 $U_i, i \in [1, n]$ 传输交换货币 c_{0i} 的预签名值 $\hat{\sigma_{0i}}$, 对应发送给 U_i; 当所有的 U_i 验证成功后, 可分别生成向 U_0 传输交换货币 c_i 的预签名值 $\hat{\sigma_i}$, 并发送给 U_0;

在交换阶段, U_0 可验证所有的传输交易和预签名值正确后, 根据证据 y, 可适配出所有对应的 SM2 签名 σ_i, 在链上公布可获得货币 c_i; 同样, U_i 通过提取算法获得证据 y, 运行适配算法 $\sigma_{0i} \leftarrow \mathsf{Adapt}(\hat{\sigma_{0i}}, y)$, 获得 U_0 向自己传输的货币 c_{0i}, 完成批量公平交换. 具体协议如图 4所示:

$U_0((X_0, x_0), X_i, c_0), i \in [1, n]$		$U_i((X_i, x_i), X_0, c_i), i \in [1, n]$
$\mathsf{GenR}(1^\lambda) \to (Y, y)$, $\mathsf{P}(Y, y) \to \pi_Y$		
$I = (Y, \pi_Y)$, $\hat{\sigma_{0i}} \leftarrow \mathsf{pSign}((X_0, x_0), I, tx_{0i})$	$\xrightarrow{I, \hat{\sigma_{0i}}, tx_{0i}}$	If $\mathsf{V}(I) \to 0$ or $\mathsf{pVrfy}(X_0, tx_{0i}, I, \hat{\sigma_{0i}}) \to 0$ Output $\perp$
If $\exists\ i \in [1, n]$, $\mathsf{pVrfy}(X_i, tx_i, I, \hat{\sigma_i}) \to 0$ Output $\perp$ else, $\sigma_i \leftarrow \mathsf{Adapt}(\hat{\sigma_i}, y)$ Publish σ_i on blockchain	$\xleftarrow{\hat{\sigma_i}, tx_i}$ $\xrightarrow{\sigma_i}$	else, $\hat{\sigma_i} \leftarrow \mathsf{pSign}((X_i, x_i), tx_i, I)$ $y \leftarrow \mathsf{Ext}(\sigma_i, \hat{\sigma_i}, I)$ $\sigma_{0i} \leftarrow \mathsf{Adapt}(\hat{\sigma_{0i}}, y)$ Publish σ_{0i} on blockchain

图 4 基于 SM2 适配器签名的批量原子交换协议
Figure 4 Batch atomic swap protocol based on SM2-based adaptor signature

批量原子交换协议可实现单参与方与多方成批次地进行跨链货币公平交换, 可高效应用于单方交易量较大的公平交换场景. 值的注意的是每批次使用相同的困难关系, 单方需要验证所有交易的正确性后才能公布真实的 SM2 签名 (任意 SM2 签名 σ_i 的公布都会泄漏证据 y).

相比于 ECDSA-AS 和 SM2-AS1, 支持离线批量证明的适配器签名 SM2-AS2 更加适用于批量原子交换协议. 在执行交换协议之前, 单方 U_0 作为困难关系选择方, 生成困难关系 (I_Y, y) 并为多方 $U_i, i \in [1, n]$ 离线批量地生成预签名公开参数和对应的零知识证明, 后续交换协议中, 多方使用验证正确的预签名公开参数能使得预签名操作更加高效. 此外, U_0 在预签名过程中, 同一批 (相同证据 y) 对应相同的预签名公开参数 $Z_0 = y(X_0 + G)$ 和零知识证明 π_{Z_0}, 因此, Z_0 和 π_{Z_0} 计算一次即可. 但是对于 ECDSA-AS, 同一批每一个预签名都对应不同的预签名公开参数 $K = k_i Y$, 导致 U_0 在给每个 U_i 计算预签名时, 需要重新选择随机数, 重新计算预签名公开参数和零知识证明, 并传输给 U_i, 计算量和通信量较大.

6.2 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. 区块链上每笔交易都需要各方 (矿工) 认可和存储是导致交易吞吐量差的关键问题, 但 PCN 可通过将部分交易转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, 27] 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 $U_1,\cdots,U_{k-1}$ 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} 的交易. 具体协议如图 5:

- 建立阶段: S 选择随机数 $l_j\in\mathbb{Z}_n$, $j=0,\cdots,k-1$, 计算 $y_j=\sum_{i=0}^{j}l_i \bmod n$ 和 $Y_j=y_jG$, 并证明 $\pi_{Y_j}\leftarrow \mathsf{P}(Y_j,y_j)$, 计算 $Z_j=y_jX_j+Y_j$, $I_j=(G,(Y_j,\pi_{Y_j}),X_j,Z_j)$, $\pi_j\leftarrow\mathsf{P}(I_j)$, $j=0,\cdots,k-1$. 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: $\mathsf{V}(I_j)\to 1$ 和 $Y_j-Y_{j-1}=l_jG$, 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 $\hat{\sigma}_0\leftarrow\mathsf{pSign}\left((X_0,x_0),tx_0,I_0\right)$ 并发送给 U_1, U_1 验证预签名值 $\mathsf{pVrfy}(X_0,tx_0,I_0,\hat{\sigma}_0)\to 1$, 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 $\hat{\sigma}_j\leftarrow\mathsf{pSign}((X_j,x_j),tx_j,I_j)$, $j=1,\cdots,k-1$ 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 $\sigma_{k-1}\leftarrow\mathsf{Adapt}(\hat{\sigma}_{k-1},y_{k-1})$ 获得 U_{k-1} 的支付. R 发送 σ_{k-1} 给 U_{k-1}; U_{k-1} 运行提取算法提取证据 $y_{k-1}\leftarrow\mathsf{Ext}(\sigma_{k-1},\hat{\sigma}_{k-1},I_{k-1})$, 计算 $y_{k-2}=y_{k-1}-l_{k-1}$, 运行适配算法计算 SM2 签名值 $\sigma_{k-2}\leftarrow\mathsf{Adapt}(\hat{\sigma}_{k-2},y_{k-2})$ 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 $y_j\leftarrow\mathsf{Ext}(\sigma_j,\hat{\sigma}_j,I_j)$, 并计算出 $y_{j-1}=y_j-l_j$, 然后通过适配算法计算 SM2 签名 $\sigma_{j-1}\leftarrow\mathsf{Adapt}(\hat{\sigma}_{j-1},y_{j-1})$ 获得 U_{j-1} 的支付, 并将 σ_{j-1} 发送给 U_{j-1}, $j=k-1,k-2,\cdots,1$. 最后, U_1 可计算 SM2 签名 σ_0 获得 S 的支付.

相比于 SM2-AS1 和 ECDSA-AS, 支持离线批量证明的适配器签名 SM2-AS2 更加适用于上述多跳支付协议, 特别是中介节点较多的支付通道网络. 发送方 S 作为困难关系选择方, 生成困难关系 (I_Y,y), 可为中介节点生成预签名阶段使用的预签名公开参数和对应的零知识证明, 避免了各中介节点在预签名阶段自己计算, 使得各中介节点预签名操作更加高效.

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

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

7 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, 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, 提出 “离线批量证明技术”, 在不影响可证明安全性的情况下, 构造更加高效的支持离线批量证明的 SM2 适配器签名方案. 该方案与现有 ECDSA 适配器签名方案相比, 预签名公开参数和对应的零知识证明可由困难关系选择方离线批量生成, 不需要预签名方在预签名阶段单独计算, 计算效率更高, 应用优势更强. 最后, 我们在现有适配器签名应用的基础上, 给出 SM2 适配器签名在区块链上的具体应用, 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. mimblewimble team mailing list. https://lists.launchpad.net/mimblewimble/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] Eckey, 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] Gugger, 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] 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

[26] 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)

[27] 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

[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] Fiat A., Shamir A. How To Prove Yourself: Practical Solutions to Identification and Signature Problems. In: Odlyzko A.M. (eds) Advances in Cryptology —CRYPTO'86. CRYPTO 1986. Lecture Notes in Computer Science, vol 263. Springer (1987), https://doi.org/10.1007/3-540-47721-7_12

[30] Chaum D., Pedersen T.P. Wallet Databases with Observers. In: Brickell E.F. (eds) Advances in Cryptology — CRYPTO'92. CRYPTO 1992. Lecture Notes in Computer Science, vol 740. Springer (1993), https://doi.org/10.1007/3-540-48071-4_7

[31] Peng C, Luo M, He D B, and Huang X Y. Adaptor Signature Scheme Based on the SM2 Digital Signature Algorithm. Journal of Computer Research and Development, 2021.
彭聪, 罗敏, 何德彪, 黄欣沂. 基于 SM2 数字签名算法的适配器签名方案 [J]. 计算机研究与发展, 2021.

[32] 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