中国科学: 信息科学 2024 年 第 54 卷 第 8 期: 1860–1870
SCIENTIA SINICA Informationis
论文

全域哈希椭圆曲线签名

张方国[1,2]

1. 中山大学计算机学院, 广州 510006
2. 广东省信息安全技术重点实验室, 广州 510006
E-mail: isszhfg@mail.sysu.edu.cn

收稿日期: 2024–2–28; 接受日期: 2024–4–11; 网络出版日期: 2024–07–30

国家重点研发计划 (批准号: 2022YFB2701500)、国家自然科学基金 (批准号: 62272491) 和广东省信息安全技术重点实验室 (批准号: 2023B1212060026) 资助项目

摘要 椭圆曲线密码体制 (elliptic curve cryptosystem, ECC) 依然是当前应用最广泛的公钥密码体制, 其安全核心是椭圆曲线离散对数问题. 本文提出了椭圆曲线离散对数的强不动点问题. 利用强不动点假设, 在随机预言模型下证明了 ECDSA (elliptic curve digital signature algorithm) 的一个全域哈希变形方案是可以抵抗自适应选择消息下的存在伪造的. 签名的聚合性质使得签名方案在诸如区块链、云存储等众多场景中发挥着重要作用, 所以本文也讨论了这个全域哈希椭圆曲线签名方案的聚合性质.

关键词 椭圆曲线, 数字签名, 不动点, 加和多项式, 聚合签名

1 引言

ECDSA (elliptic curve digital signature algorithm) [1] 是 DSA [2] (digital signature algorithm) 的椭圆曲线变形, 是一种基于椭圆曲线离散对数问题设计的数字签字算法. ECDSA 最早是由 Vanstone 等 [1] 在 1992 年为了响应 NIST 对数字签名标准公众评价的要求而提出的. 该签名算法于 1998 年作为 ISO 标准被采纳, 在 1999 年作为 ANS 标准被采纳, 并于 2000 年成为 IEEE 和 FIPS 标准. 从居民身份证到密码货币, ECDSA 在当前信息时代具有广泛的应用.

最早利用离散对数问题设计的数字签名是 ElGamal 签名方案 [3], 1991 年由德国密码学家 Schnorr [4] 提出的 Schnorr 签名方案可以看作 ElGamal 签名方案的一种变形, 它缩短了签名长度. DSA [2] 是 1991 年 NIST 提出的数字签名标准 DSS 中所使用的签名算法, 它是 ElGamal 签名方案的另一种变形, 同时也吸收了 Schnorr 方案的一些设计思想. ElGamal, Schnorr, DSA 三种签名方案都可归结为基于有限域的离散对数签名体制的特例. 离散对数签名体制的签名等式和验证等式有多种构造方式, 不同的构造方法得出不同的签名算法, 所以实际上基于离散对数问题设计的数字签名有很多种变形. Horster 等 [5] 在 1994 年研究了基于离散对数问题的签名方案的构造, 提出了 Meta-ElGamal

引用格式: 张方国. 全域哈希椭圆曲线签名. 中国科学: 信息科学, 2024, 54: 1860–1870, doi: 10.1360/SSI-2024-0064
Zhang F G. Full domain Hash elliptic curve signature (in Chinese). Sci Sin Inform, 2024, 54: 1860–1870, doi: 10.1360/SSI-2024-0064

签名 (即变形的或多样化的 ElGamal 签名). 这些变形方案自然可以推广到椭圆曲线情形. 很多基于离散对数问题的签名方案都是 Meta-ElGamal 签名的某个实例. 这些变形方案在安全性上基本是等价的. 这么多等价变形签名方案, 选择哪一种使用, 主要是考量所选的方案在实现效率上是否有优势, 或在功能上是否有特色.

一个签名方案如果具有批验证或聚合性质, 在很多场景具有重要应用, 例如在云存储或区块链应用中, 经常会用到同时验证多个签名或压缩存储多个签名的情形. 批验证是一种特殊的数字签名验证方法, 它允许验证者一次验证多个签名. 聚合签名方案允许为 l 个不同签名者对不同消息的 l 个签名创建一个紧致的签名, 其主要设计目标是将多个签名数据压缩合并成单个聚合签名. 这就提供了更快的验证速度和更少的存储空间, 以及传输带宽上的节省. 聚合签名可以有效降低存储空间和验证过程中网络流量成本, 尤其对签名频次较低但验证频次较高的业务场景有显著效果.

2001 年, Zhang 等 [6] 提出了 3 个 ECDSA 的变形方案, 其中有一个方案 (文献 [6] 中的方案三) 是首先将要签署的消息哈希到椭圆曲线群, 有点类似全域哈希签名. 这样的好处是可以对消息预处理, 同时签名具有部分聚合性质. 本文重新考虑这个 ECDSA 的变形方案, 引入椭圆曲线离散对数的强不动点假设, 并在该假设下, 在随机预言模型下证明了这个签名方案是安全的. 我们也进一步讨论了这个方案的聚合性质.

2 预备知识

2.1 安全签名

数字签名的定义可如下给出.

定义1 (数字签名方案) 一个数字签名方案由 3 个多项式时间算法 (KeyGen, Sign, Verify) 组成.

• KeyGen: 密钥生成算法, 以一个安全参数 1^k 作为输入, 输出一对密钥 (pk, sk), 称为签名公钥以及私钥.

• Sign: 签名算法, 以一个私钥 sk 以及一个消息 m 作为输入, 然后输出一个签名 σ, 记作

$$\sigma \leftarrow \mathrm{Sign}_{\mathrm{sk}}(m).$$

• Verify: 验证算法, 一个确定性函数, 以一个公钥 pk 以及一个消息签名对 (m, σ) 作为输入, 如果

$$\mathrm{Verify}_{\mathrm{pk}}(m, \sigma) = 1,$$

则称 σ 是对消息 m 的有效签名.

有时也将签名所用到的参数利用一个参数生成算法 ParamGen 来表示, 并一起加到数字签名的定义中. 所以一般一个签名方案可以用 $\mathcal{S} = \langle \mathsf{ParamGen}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify} \rangle$ 来表示.

任何一个密码原语的安全性定义都需要考虑两个方面: 一方面是敌手最强的攻击手段; 另一方面就是敌手最基本 (或最低的) 攻击目标. 对于一个签名方案来说, 敌手最强的攻击手段是自适应选择消息攻击, 而敌手的最基本的攻击目标是存在伪造. 所以数字签名方案的安全性定义就是在自适应选择消息攻击下不可存在伪造.

大多数的签名方案会用到哈希函数, 即先哈希后签名. 当哈希函数的值域与用来构造签名方案的单向陷门函数的定义域相同时, 我们称这样的签名为全域哈希签名. 在安全证明时, 我们通常把此时的哈希函数看成是一个随机预言机, 即哈希函数的输出是值域中一个随机分布的元素.

我们也可以利用 $t - \epsilon$ 语言给出先哈希后签名的签名方案的精确安全性的定义.

定义2 (签名的精确安全性 [7]) 一个伪造者 $\mathcal{F}$ 被称为是用自适应选择消息攻击 (t, q_H, q_S, ϵ)- 攻破签名方案 $\mathcal{S} = \langle \mathsf{ParamGen}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify} \rangle$ 的, 如果它至多进行了 q_H 次哈希预言机询问, q_S 次签名询问和用了处理时间 t 后, 以至少是 ϵ 的概率输出一个合法伪造.

如果不存在伪造者能够 (t, q_H, q_S, ϵ) 攻破一个签名方案 $\mathcal{S}$, 则我们称这个签名方案是 (t, q_H, q_S, ϵ)-安全的.

2.2 有限域上椭圆曲线

设 $\mathbb{F}_q$ 是一个含有 q 个元素的有限域. 由 Weierstrass 方程确定的光滑 (或非奇异) 曲线

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

加上无穷远点 $\mathcal{O}$ 被称为有限域 $\mathbb{F}_q$ 上的椭圆曲线, 记为 $E(\mathbb{F}_q)$, 其中 $a_1, \ldots, a_6 \in \mathbb{F}_q$ 为常数, 即

$$E(\mathbb{F}_q) = \{\mathcal{O}\} \cup \{(x, y) \in \mathbb{F}_q \times \mathbb{F}_q \,|\, y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6, a_i \in \mathbb{F}_q\}.$$

如果有限域 $\mathbb{F}_q$ 的特征既不是 2 也不是 3, 那么椭圆曲线方程可以写成

$$y^2 = x^3 + ax + b,$$

其中 $a, b \in \mathbb{F}_q$ 为常数, 且满足 $4a^3 + 27b^2 \neq 0$.

在借助切割线法则定义了椭圆曲线 $E(\mathbb{F}_q)$ 上有理点的加法运算后, 椭圆曲线 $E(\mathbb{F}_q)$ 构成一个加法交换群, 并以无穷远点为群单位元 [8]. 这个群要么是一个循环群, 要么就是两个循环群的直积.

Hasse 研究了有限域上椭圆曲线有理点的个数和 Frobenius 变换的迹的关系, 给出如下结论.

定理1 (Hasse) E 是定义在有限域 $\mathbb{F}_q$ 上的椭圆曲线, $t = q + 1 - \sharp E(\mathbb{F}_q)$, 则 $|t| \leqslant 2\sqrt{q}$.

也就是说, Hasse 定理给出了 $\mathbb{F}_q$ 上椭圆曲线的点数的一个范围, 即位于 $[q+1-2\sqrt{q}, q+1+2\sqrt{q}]$. 关于椭圆曲线的更多理论和算法, 可以参看文献 [8, 9].

定义3 (elliptic curve discrete logarithm problem, ECDLP) 设 $P \in E(\mathbb{F}_q)$ 且 $\langle P \rangle$ 是由点 P 生成的加法子群. 如果点 $Q \in \langle P \rangle$, 则存在某一整数 k 满足 $kP = Q$. 计算这样的整数 k 的问题被称为椭圆曲线离散对数问题 (ECDLP).

假设点 P 的阶 $\mathrm{ord}(P) = n$, 那么 k 实际上是模 n 的剩余类. 与一般对数记号类似, 可以把椭圆曲线离散对数记为

$$\log_P Q \equiv k \bmod n.$$

我们考虑定义在大素数域上的椭圆曲线, 这是密码应用中最常见的一类曲线. 记 $E(\mathbb{F}_p)$ 为大素数域 $\mathbb{F}_p$ 上的一条椭圆曲线. 文献 [10] 推广了模素数 p 离散对数的不动点问题, 给出了椭圆曲线 $E(\mathbb{F}_p)$ 上 n 阶子群中离散对数的不动点问题, 具体定义如下.

定义4 (ECDLP 不动点) 设 $P \in E(\mathbb{F}_p)$ 且 $\mathrm{ord}(P) = n$. $\langle P \rangle \subseteq E(\mathbb{F}_p)$ 上关于 P 的离散对数的一个不动点是一个满足 $\log_P Q = x$ 的点 $Q = (x, y) \in \langle P \rangle$, 也即 $xP = Q$, 其中 x 被看成是区间 $[0, n-1]$ 上的一个整数.

文献 [10] 同时证明了如果 p 和 n 足够大, 对于 $\mathbb{F}_p$ 上的任意椭圆曲线, 以大概率存在阶为 n 的点 Q 是 $\langle P \rangle \subseteq E(\mathbb{F}_p)$ 上以底数 P 为离散对数的不动点.

2.3 椭圆曲线的加和多项式

2004 年, Semaev [11] 引进了椭圆曲线的加和多项式 (summation polynomials) 实现点分解, 用以构造有理点之间的关系, 从而可以利用指标计算的框架求解 ECDLP.

我们考虑定义在大素数有限域 $\mathbb{F}_p$ 上的一条椭圆曲线

$$E: y^2 = x^3 + ax + b.$$

定义 E 的第 2 和 3 个加和多项式分别为

$$S_2(x_1, x_2) = x_1 - x_2,$$

$$S_3(x_1, x_2, x_3) = (x_1 - x_2)^2 x_3^2 - 2((x_1 + x_2)(x_1 x_2 + a) + 2b)x_3 + ((x_1 x_2 - a)^2 - 4b(x_1 + x_2)).$$

对于任意第 $r > 3$ 个加和多项式, 我们可以借助结式, 利用递归的方式定义

$$S_r(x_1, \ldots, x_r) = \mathrm{Res}_x(S_{r-1}(x_1, \ldots, x_{r-2}, x), S_3(x_{r-1}, x_r, x)).$$

我们也可以推广这个构造, 得出

$$S_{r+k}(x_1, \ldots, x_{r+k}) = \mathrm{Res}_x(S_{r+1}(x_1, \ldots, x_r, x), S_{k+1}(x_{r+1}, x_{r+k}, x)).$$

通过定义可以看出, 对于每个自然数 $r \geqslant 3$, E 的第 r 个加和多项式 S_r 关于每个变量 x_i 的次数是 2^{r-2}.

定义在任意域上的椭圆曲线都可以定义加和多项式, 加和多项式具有下列重要的性质.

定理2 ([11]) 令 $\overline{\mathbb{F}}$ 是域 $\mathbb{F}$ 的代数闭域. E 是 $\mathbb{F}$ 上具有系数 $A = (a_1, a_2, a_3, a_4, a_6)$ 的椭圆曲线. 对任意的正整数 $r \geqslant 2$, 如果存在 $P_1, P_2, \ldots, P_r \in E(\overline{\mathbb{F}}) \setminus \mathcal{O}$, $P_i = (x_{P_i}, y_{P_i})$ 使得

$$P_1 + P_2 + \cdots + P_r = \mathcal{O}$$

当且仅当

$$S_{A,r}(x_{P_1}, \ldots, x_{P_r}) = 0.$$

其中 $S_{A,r}(x_1, \ldots, x_r)$ 是由系数 $A = (a_1, a_2, a_3, a_4, a_6)$ 定义的域 $\mathbb{F}$ 上的椭圆曲线 E 的第 r 个加和多项式.

3 签名方案描述

基于椭圆曲线的密码体制所依赖的单向陷门函数大都是定义在椭圆曲线群上的, 有些方案中所涉及的哈希函数的值域需要定义在椭圆曲线群上. 在文献 [6] 提出的 ECDSA 的 3 个变形方案中, 有一个方案就是需要将要签名的消息 m 通过一个消息编码方案嵌入到椭圆曲线的一个有理点上. 这使得这一方案中所使用的哈希函数与 ECDSA 及其他变形中所用的普通哈希函数不一样. 我们把这个使用了特殊哈希函数的 ECDSA 的变形方案称为全域哈希椭圆曲线签名, 记为 FDH-ECS.

尽管全域哈希椭圆曲线签名方案可以利用定义在任意有限域上的椭圆曲线来构造, 我们这里主要考虑在素数域上的实现. 系统的基本参数如下: $(\mathbb{F}_p, E, n, P)$, 这里 p, n 是素数, E 是定义在有限

算法 1 FDH-ECS 签名

输入: 待签名的消息 m 和私钥 d_A.
输出: 关于消息 m 的签名 (r,s).
1: 通过哈希函数 Hash 将消息 m 映射成 $E(\mathbb{F}_p)$ 上的一个有理点, 即 $M=\text{Hash}(m)\in E(\mathbb{F}_p)$;
2: 随机选择一个整数 $k\in\mathbb{Z}_n$;
3: 计算 $R=M+kP=(x_R,y_R)$;
4: 计算 $r=x_R \mod n$, 如果 $r=0$, 则返回到步骤 2;
5: 计算 $s=k-rd_A \mod n$, 如果 $s=0$, 则返回到步骤 2;
6: 输出 (r,s).

算法 2 FDH-ECS 验证

输入: 消息 m, 公钥 P_A 和签名 (r,s).
输出: 接受或拒绝.
1: 验证 r,s 是 $\mathbb{Z}_n$ 中的整数. 如果不是, 直接输出 "拒绝";
2: 计算 $M=\text{Hash}(m)$;
3: 计算 $R=M+sP+rP_A=(x_R,y_R)$, 如果 $R=\mathcal{O}$, 则拒绝这个签名;
4: 计算 $v=x_R \mod n$;
5: 当且仅当 $v=r$ 时接受这个签名.

域 $\mathbb{F}_p$ 上的椭圆曲线, $P\in E(\mathbb{F}_p)$ 是椭圆曲线上一个 n 阶点. 另外, 还需要一个安全的哈希函数 Hash : $\{0,1\}^*\to E(\mathbb{F}_p)$.

假定用户的公私钥对是 $(P_A=d_AP,d_A)$. FDH-ECS 的签名生成算法如算法 1 所示.

验证者验证 (r,s) 是否是签名者对消息 m 的签名, 如算法 2 所示.

签名验证的正确性:

$$\text{Hash}(m)+sP+rP_A=M+(k-rd_A)P+rP_A=M+kP=R=(x_R,y_R),\ r=x_R \mod n.$$

全域哈希椭圆曲线签名需要用到一个特殊的哈希函数 Hash : $\{0,1\}^*\to E(\mathbb{F}_p)$, 这样的哈希函数也称为消息嵌入编码. 这一函数可以首先通过一个一般哈希函数 $h:\{0,1\}^*\to\mathbb{F}_p$ 将要签名的消息 m 映射到有限域 $\mathbb{F}_p$ 上, 然后将 h 的这一输出值当成椭圆曲线某一有理点的 x 坐标. 这样的哈希算法适用于所有有限域上的椭圆曲线, 但这是一个概率算法, 将任意消息映射到椭圆曲线上的有理点的成功概率是 1/2.

第一个确定性的多项式时间椭圆曲线嵌入算法是 Shallue 和 Woestijne [12] 在 ANTS 2006 上提出的, 之后被 Ulas [13] 进行了简化, 并推广到超椭圆曲线情形, 简称为 SWU 算法. 这一算法在任意有限域 $\mathbb{F}_p$ 上的运行时间是 $O(\log^4 p)$. 当 $p\equiv 3 \mod 4$ 时, 这一算法的运行时间可降低为 $O(\log^3 p)$. 在 2009 年 International Cryptology Conference (CRYPTO) 上, Icart [14] 针对 $p\equiv 2 \mod 3$ 的有限域 $\mathbb{F}_p$ 提出了一个计算复杂度为 $O(\log^3 p)$ 的确定性算法. 令 $f:\mathbb{F}_p\to E(\mathbb{F}_p)$ 是由 SWU 或 Icart 算法定义的函数, 则可定义全域哈希函数为 $\text{Hash}(m)=f(h(m))$. 如果只是将此 Hash 当成消息压缩函数, 这样的构造是可以的. 但如果将此 Hash 看成随机预言机, 这样的构造还不满足随机预言机的要求, 因为这样定义的全域哈希在 f 函数的像集中的分布不是均匀的.

Brier 等 [15] 研究了 SWU 和 Icart 算法, 给出了到有限域 $\mathbb{F}_p$ 上阶为 n 的椭圆曲线群 $E(\mathbb{F}_p)=\langle P\rangle$ 上的与随机预言机不可区分的全域哈希函数的一般构造. 特别地, 当 f 函数是由 Icart 算法定义时 (即

有限域的特征满足 $p \equiv 2 \mod 3$), Brier 等给出了如下的有效的全域哈希函数的构造:

$$\mathrm{Hash}(m) := f(h_1(m)) + f(h_2(m)),$$

其中 $h_1 : \{0,1\}^* \to \mathbb{F}_p$ 和 $h_2 : \{0,1\}^* \to \mathbb{F}_p$ 是两个哈希函数. 并证明了当 h_1, h_2 被看成是随机预言机时, 此 $\mathrm{Hash}(m)$ 和随机预言是不可区分的.

4 签名方案的安全性

4.1 强不动点假设

下面给出椭圆曲线离散对数不动点问题的一个变形, 称为强不动点问题.

定义5 (强不动点问题) 设 $P \in E(\mathbb{F}_p)$ 且 $\mathrm{ord}(P) = n$. 给定 P 和 $Q \in \langle P\rangle \subseteq E(\mathbb{F}_p)$, 计算 $s, t \in \mathbb{Z}_n$, 使得

$$sP - tQ = (t, y) \in E(\mathbb{F}_p).$$

很显然, 如果 ECDLP 可解, 则强不动点问题可解. 简单证明如下: 给定强不动点问题实例 $(E(\mathbb{F}_p), n, P, Q)$, 随机选择 $t \in \mathbb{Z}_n$, 使得 t 是 $E(\mathbb{F}_p)$ 上一有理点的 x 坐标, 计算 $tQ + (t, y)$ 关于 P 的离散对数, 记为 s, 即 $s = \log_P(tQ + (t, y))$, 则此时 s, t 就是强不动点问题 $(E(\mathbb{F}_p), n, P, Q)$ 的一个解.

现在还不确定椭圆曲线的强不动点问题是否等价于 ECDLP, 这是一个值得继续深入研究的问题.

定义6 (强不动点假设) $\langle P\rangle \subseteq E(\mathbb{F}_p)$ 上离散对数强不动点问题是困难的, 即没有多项式时间算法能以不可忽略的优势解决强不动点问题.

我们也可以用 (t, ϵ)- 语言描述强不动点假设: $\langle P\rangle \subseteq E(\mathbb{F}_p)$ 上离散对数强不动点问题是 (t, ϵ)- 安全的, 即不存在算法能在 t 时间内以 ϵ 的成功概率求解强不动点问题.

4.2 安全证明

我们在强不动点假设下证明全域哈希签名方案是安全的.

定理3 假定椭圆曲线强不动点问题是 (t', ϵ')- 安全的. 那么 FDH-ECS 方案是 (t, ϵ)- 安全的, 其中

$$t = t' - (q_{\mathrm{hash}} + q_{\mathrm{sig}}) \cdot O(\log^3 n),\ \epsilon = \exp(1) q_{\mathrm{sig}} \epsilon'.$$

方案的安全证明类似于全域哈希的 RSA 方案的证明 [7].

证明 假定对任意的全域哈希椭圆曲线签名方案 FDH-ECS, 存在一个伪造者 $\mathcal{F}$ 可以 $(t, q_{\mathrm{sig}}, q_{\mathrm{hash}}, \epsilon)$ 攻破它. 那么我们利用 $\mathcal{F}$ 构造一个算法 $\mathcal{A}$ 求解椭圆曲线离散对数的强不动点问题.

假定要求解的椭圆曲线离散对数的强不动点问题如下:

设 $P \in E(\mathbb{F}_p)$ 且 $\mathrm{ord}(P) = n$. 给定 P 和 $Q \in \langle P\rangle \subseteq E(\mathbb{F}_p)$, 计算 $s, t \in \mathbb{Z}_n$, 使得

$$sP - tQ = (t, y) \in E(\mathbb{F}_p).$$

首先利用上面强不动点问题的参数构造 FDH-ECS 方案的基本参数为 $(\mathbb{F}_q, E, n, P)$, 令 Q 为签名公钥.

当伪造者询问哈希预言机时, $\mathcal{A}$ 将扮演该预言机回答. 当 $\mathcal{F}$ 对消息 m 进行哈希询问时, $\mathcal{A}$ 累计计数 i, 令 $m_i = m$, 随机选取 $s_i, k_i \in \mathbb{Z}_n$, 计算 $Q_i = k_i Q = (x_i, y_i)$. 然后以概率 ρ 返回 $H_i = k_i Q - (x_i \mod n)Q - s_i P$, 以概率 $1 - \rho$ 返回 $H_i = k_i P$.

当 $\mathcal{F}$ 对消息 m 询问签名时, 此时 m 的哈希值为 H_i. 如果 $H_i = k_iQ + (x_i \mod n)Q - s_iP$, 则 $\mathcal{A}$ 返回 $(r_i = x_i \mod n, s_i)$ 作为消息 m 的签名, 否则终止过程, $\mathcal{A}$ 宣布失败.

不难验证, 在 $\mathrm{Hash}(m) = H_i = k_iQ + (x_i \mod n)Q - s_iP$ 时, $(r_i = x_i \mod n, s_i)$ 是消息 m 在公钥 (P, Q) 下的合法签名:

$$\mathrm{Hash}(m) + s_iP + r_iQ = k_iQ - (x_i \mod n)Q - s_iP + s_iP + r_iQ = k_iQ = (x_i, y_i).$$

最后, $\mathcal{F}$ 输出一个伪造的消息签名对 (m', r', s'). 如果此时 $\mathrm{Hash}(m') = H_j$, 且 $H_j = k_jP$, 则 $\mathcal{A}$ 就求解了椭圆曲线强不动点问题, 因为

$$\mathrm{Hash}(m) + s'P + r'Q = k_jP + sP + r'Q = (r', y),$$

所以 $s = k_j + s', t = r'$ 就是椭圆曲线强不动点 (P, Q) 的解.

假定 $\mathcal{F}$ 至多问了 q_{hash} 次全域哈希, 至多 q_{sig} 次签名询问. 则 $\mathcal{A}$ 能够回答 $\mathcal{F}$ 的所有签名询问的概率至少是 $\rho^{q_{\mathrm{sig}}}$. 这也是最后 $\mathcal{F}$ 能输出一个伪造签名的概率. 输出的这个伪造的签名所对应的全域哈希形如 $H_i = k_iP$ 的概率是 $1 - \rho$. 所以 $\mathcal{A}$ 能成功求解强不动点问题的成功概率至少是 $\alpha(\rho) = \rho^{q_{\mathrm{sig}}} \cdot (1 - \rho) \cdot \epsilon'$.

函数 $\alpha(\rho)$ 在 $\rho_{\max} = 1 - 1/(q_{\mathrm{sig}} + 1)$ 时取到最大值, 此时最大值为

$$\alpha(\rho_{\max}) = \frac{1}{q_{\mathrm{sig}}} \left(1 - \frac{1}{q_{\mathrm{sig}} + 1}\right)^{q_{\mathrm{sig}}+1} \cdot \epsilon'.$$

所以

$$\epsilon = \frac{1}{(1 - \frac{1}{q_{\mathrm{sig}}+1})^{q_{\mathrm{sig}}+1}} \cdot q_{\mathrm{sig}} \cdot \epsilon'.$$

当 q_{sig} 充分大时, 有 $\epsilon = \exp(1) q_{\mathrm{sig}} \epsilon'$.

算法 $\mathcal{A}$ 的运行时间是 $\mathcal{F}$ 的运行时间加上回答 q_{hash} 次哈希询问和 q_{sig} 次签名询问的计算量. 每次哈希的回答需要计算 3 个或 1 个标称乘, 归结到比特级别的计算量的话, 至多是 $O(\log^3 n)$, 而签名的回答只是查表和模运算, 至多是 $O(\log^2 n)$ 的比特运算. 所以 $t = t' - (q_{\mathrm{hash}} + q_{\mathrm{sig}}) \cdot O(\log^3 n)$.

5 可聚合性讨论

5.1 参数选取

在我们的签名方案中, 如果要求 $n > p$, 则在签名算法和验证算法中都可以省去 $x_R \mod n$ 这一步 (即算法中的第 4 步), 从而 $t = x_R$. 所以, 如果在椭圆曲线密码体制的基本参数选择上要求 $n > p$, 不仅可以简化全域哈希椭圆曲线签名方案的签名算法和验证算法, 还可以为后面即将讨论的签名的聚合性质提供便利.

从 Hasse 界可知, 在素数域上, 椭圆曲线群的阶可以大于域的大小. 给定有限域, 从安全性方面, 尽可能地希望所得到的群的阶大一些. 群的阶越大, 离散对数计算越困难. 然而在我们所看到的大多数素数域上的椭圆曲线密码体制的标准中, 所用到的群的阶大都是小于有限域的大小的, 例如 NIST 曲线、SM2 建议曲线、SEC 标准曲线以及 IEEE P1363 建议的曲线等. 为什么会是这样的呢? 实际上出现这个现象主要是从实现密码方案的有效性考虑的. 在大多数的椭圆曲线密码标准中, 用来定义有限域的素数大都是广义梅森 (Mersenne) 素数 (形如 $p = 2^\lambda - c$), 并且 λ 一般取计算机的处理器尺寸的

倍数, 例如都是 32 的倍数 (如 NIST P-192: $p = 2^{192} - 2^{64} - 1$, SEC 的 secp256k1: $p = 2^{256} - 2^{32} - 977$ 等). 因为是利用广义梅森素数, 此时有限域的大小已经接近 λ 位, 尽管我们可以随机选曲线, 但当 c 比较小时, 会使得选取的椭圆曲线在有限域上的有理点数超过 2^{λ}, 即 n 会大于 2^{λ}, 此时的签名算法中的模 n 运算就会是一个超出 λ 比特的模运算, 从而计算效率会降低. 所以一般选取安全椭圆曲线时, 椭圆曲线的阶 n 选得要比 p 小, 这样模 n 的处理就不会超出 λ 位. 当 p 的值与 2^{λ} 差距很大时, 我们也可以选取 $n > p$, 使得模 n 的处理不会超出 λ 位, 并且选取空间很大.

例如考虑 NIST P-256 和 Secp256r 所用的素数域 $\mathbb{F}_p$, 其中

$$p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1.$$

显然, p 与 2^{256} 的差距超过了 $2\sqrt{p}$, 所以可以随机在 $\mathbb{F}_p$ 上选取曲线, 得到的曲线的阶不会超出 2^{256}. 例如我们可以考虑如下定义的椭圆曲线 $y^2 = x^3 + ax + b$, 这里 $a = -3$, b 取

76852578556138553424544674993256502483760026821577798839125201069695503033348.

此时椭圆曲线群的阶 n 是如下的素数, 且大于 p:

115792089210356248762697446949407573530223359421938169267226143902323772114679.

5.2 半聚合签名

在签名生成算法中, 将多个用户对多个消息的独立签名聚合压缩成单个签名, 并且保证压缩过程对总体签名尺寸做了减少; 而且, 如果最终的聚合签名验证有效, 则意味着原始所有用户的单个签名都有效. 带有这种聚合性质的签名方案具有批量验证的优势, 这使得这类具有聚合性质的签名方案以其特有的优势被广泛应用于无线传感网、证书链认证、安全路由等领域当中.

聚合签名在 2003 年由 Boneh 等 [16] 首次定义, 通过将数字签名压缩技术和批处理技术进行整合, 设计了第一个基于双线性对的方案. 基于离散对数问题设计的签名方案, 例如 Schnorr 签名, DSA (ECDSA) 以及它们的变形方案, 签名都是包含两个元素的. 在这类签名的聚合中, 将两部分都压缩是很难构造的, 一般只能聚合一部分, 所以这类基于离散对数签名的聚合都是半聚合签名, 例如 Chen 等 [17] 给出的 Schnorr 签名的半聚合方案以及 Chalkias 等 [18] 给出的 EdDSA 和 Schnorr 变形签名的半聚合方案等.

聚合签名可如下形式化定义.

定义7 (聚合签名方案) 聚合签名方案包含 5 个算法 (KeyGen, Sign, Verify, AggregateSign, AggregateVerify), 前 3 个与普通签名方案一样.

• KeyGen: 密钥生成算法, 以一个安全参数 1^k 作为输入, 输出一对密钥 $(\mathrm{pk}, \mathrm{sk})$, 称为签名公钥以及私钥.

• Sign: 签名算法, 以一个私钥 sk 以及一个消息 m 作为输入, 然后输出一个签名 σ.

• Verify: 验证算法, 以一个公钥 pk 以及一个消息签名对 (m, σ) 作为输入的确定性函数, 输出是接受或拒绝.

• AggregateSign: 签名聚合算法, 输入 l 个公钥、消息签名组 $((\mathrm{pk}_1, m_1, \sigma_1), \ldots, (\mathrm{pk}_l, m_l, \sigma_l))$, 输出一个聚合签名 σ_{aggr}.

• AggregateVerify: 聚合签名的验证算法, 对输入的一组公钥和消息 $((\mathrm{pk}_1, m_1), \ldots, (\mathrm{pk}_l, m_l))$ 以及一个聚合签名 σ_{aggr}, 输出是接受或拒绝.

只要得到的聚合签名的长度比参与聚合的所有签名的长度和小, 都算是一个成功的聚合签名. 有的聚合签名的长度和单个签名长度相当, 这是很完美的聚合签名. 有些聚合签名方案的聚合签名长度

只达到了所有参与聚合签名的长度和的一半, 或比一半略多一点, 则称这类聚合签名为半聚合或半 $+\epsilon$ 聚合 [18].

我们的全域哈希椭圆曲线签名方案具有半聚合性质. 为了便于聚合验证, 我们假定 $n > p$, 此时 FDH-ECS 中的 r 就不需要模 n, 仍然是 $\mathbb{F}_p$ 中元素.

给定有 l 个公钥、消息和签名元组 $(P_i, m_i, (r_i, s_i))$, 我们可以把签名中的 s_i 合成一个元素, 即 $s = \sum s_i \mod n$, 从而使得签名节省了将近一半的存储空间. 此时的聚合签名为 $(r_1, r_2, \ldots, r_l, s)$. 对于聚合后的签名, 验证者首先计算

$$\mathrm{Hash}(m_1) + \cdots + \mathrm{Hash}(m_l) + sP + r_1P_1 + \cdots + r_lP_l = (x_R, y_R).$$

然后验证是否有 $S_{l+1}(r_1, r_2, \ldots, r_l, x_R) = 0$. 这里 S_{l+1} 是椭圆曲线的第 $l+1$ 个加和多项式.

在同一公钥下的多个签名的部分聚合如下: 给定公钥 (P, P_A), 以及在此公钥下的 l 个消息签名对 $(m_i, (r_i, s_i))$, 我们将这些消息签名对聚合如下: $(m_1, \ldots, m_l), (r_1, \ldots, r_l, s = \sum s_i)$.

验证者首先计算

$$\mathrm{Hash}(m_1) + \cdots + \mathrm{Hash}(m_l) + sP + (r_1 + \cdots + r_l)P_A = (x_R, y_R).$$

然后验证是否有 $S_{l+1}(r_1, r_2, \ldots, r_l, x_R) = 0$.

聚合签名验证的正确性推导如下: 对每个 i, $(m_i, (r_i, s_i))$ 是公钥 P_i 下的合法的 FDH-ECS 消息签名对, 且由于 $n > p$, 则有

$$\mathrm{Hash}(m_i) + s_iP + r_iP_i = (r_i, y_i).$$

而

$$\mathrm{Hash}(m_1) + \cdots + \mathrm{Hash}(m_l) + sP + r_1P_1 + \cdots + r_lP_l = \sum_{i=1}^{l}(\mathrm{Hash}(m_i) + s_iP + r_iP_i) = \sum_{i=1}^{l}(r_i, y_i),$$

从而有

$$\sum_{i=1}^{l}(r_i, y_i) = (x_R, y_R),$$

即

$$(r_1, y_1) + \cdots + (r_l, y_l) + (x_R, -y_R) = \mathcal{O}.$$

利用椭圆曲线加和多项式的性质 (定理 2) 可得 $S_{l+1}(r_1, r_2, \ldots, r_l, x_R) = 0$.

如果一个敌手没有消息 $m_1, m_2, \ldots, m_l$ 的合法签名, 而想去伪造一个聚合签名 $(r'_1, \ldots, r'_l, s')$, 这等价于去求解下列问题的一组解 $(r'_1, \ldots, r'_l, s')$:

$$\mathrm{Hash}(m_1) + \cdots + \mathrm{Hash}(m_l) + s'P + r'_1P_1 + \cdots + r'_lP_l = (r'_1, y_1) + \cdots + (r'_l, y_l).$$

这个问题就相当于去求解一个推广的强不动点问题.

我们的半聚合椭圆曲线签名比原来的 l 个签名的长度减少了将近一半. 下面讨论一下聚合签名的验证效率. 对于定义在大素数域的椭圆曲线点加运算在仿射坐标下的代价是 $1I + 1S + 2M$, 在混合坐标下 (Chudnovsky 射影坐标 + 仿射坐标) 可以达到 $3S + 8M$ 的计算量; 倍点运算在仿射坐标下是 $1I + 2S + 2M$, 在如 Jacobian 射影坐标下可以达到 $4S + 4M$ 的计算量. 这里 M, S, I 分别记为域 $\mathbb{F}_p$

上的一次乘法、一次平方和一次逆运算. 令 $\lambda = \lceil \log_2 n \rceil$, 按照文献 [19] 第 3.3 小节的分析, 一次标量乘的计算量大约是 $8\lambda M + 5.5\lambda S + (1I + 3M + 1S)$. 由于求逆运算的计算量大于乘法, 所以标量乘的计算一般在射影坐标下进行, 最后的 1 次求逆, 3 次乘法和 1 次平方是将射影坐标返回到仿射坐标的代价.

不考虑消息嵌入的哈希运算, l 个 FDH-ECS 消息签名对的验证需要 $l \times (2A + 2\mathrm{PM})$ 次运算. 这里 A 表示椭圆曲线的点加运算, 而 PM 表示椭圆曲线的标量乘. 将 l 个 FDH-ECS 消息签名对的验证转化成域 $\mathbb{F}_p$ 上的运算代价为

$$l \times (16\lambda M + 11\lambda S + (1I + 19M + 7S)).$$

对于聚合签名的验证, 不同公钥情形的验证代价是

$$2l \times A + (l + 1) \times \mathrm{PM} + 1\mathrm{SP}(l + 1),$$

相同公钥情形的验证代价是

$$(l + 1) \times A + 2 \times \mathrm{PM} + 1\mathrm{SP}(l + 1),$$

其中 $\mathrm{SP}(l + 1)$ 表示椭圆曲线的第 $l + 1$ 个加和多项式的估值计算. $S_{l+1}(r_1, r_2, \ldots, r_l, x_R)$ 的计算至多需要 2^{l^2} 次 $\mathbb{F}_p$ 上的乘法运算. 统一考虑成域 $\mathbb{F}_p$ 上的运算, 则不同公钥情形和相同公钥情形的聚合签名的验证代价分别是 $2l \times (8M + 3S) + (l + 1) \times (8\lambda M + 5.5\lambda S) + (1I + 3M + 1S) + 2^{l^2} M$ 和 $2l \times (8M + 3S) + 2 \times (8\lambda M + 5.5\lambda S) + (1I + 3M + 1S) + 2^{l^2} M$.

例如考虑 Secp256r 所用的素数域上的 FDH-ECS 的实现, 同一公钥下的两个签名的验证的计算代价是 $2 \times (2A + 2\mathrm{PM}) = 8198M + 5646S + 2I$. 将这两个签名聚合后, 验证所需要的有限域运算量为 $3A + 2\mathrm{PM} + 1\mathrm{SP}(3) = 4128M + 2829S + 1I$. 这里第 3 个加和多项式的计算只需要有限域中的 3 个平方和 5 个乘法.

如果考虑加和多项式的结式生成方式, 并且只是检验解的话, $S_{l+1}(r_1, r_2, \ldots, r_l, x_R) = 0$ 的计算应该会比 2^{l^2} 少, 从而使得聚合验证更有效.

6 结束语

我们提出了椭圆曲线离散对数的强不动点问题. 利用强不动点假设, 在随机预言模型下证明了 ECDSA 的一个全域哈希变形方案 FDH-ECS 是可以抵抗自适应选择消息下的存在伪造的. 签名的聚合性质在区块链等场景具有重要应用, 所以本文还讨论了 ECDSA 的变形方案的聚合性.

参考文献

1 Johnson D, Menezes A, Vanstone S. The elliptic curve digital signature algorithm (ECDSA). Int J Inf Sec, 2001, 1: 36–63

2 National Institute of Standards and Technology (NIST). FIPS publication 186: Digital Signature Standard (DSS). 1994. https://csrc.nist.gov/pubs/fips/186/final

3 Elgamal T. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans Inform Theor, 1985, 31: 469–472

4 Schnorr C P. Efficient signature generation by smart cards. J Cryptology, 1991, 4: 161–174

5 Horster P, Petersen H, Michels M. Meta-ElGamal signature schemes. In: Proceedings of the 2nd ACM Conference on Computer and Communications Security, 1994. 96–107

6 Zhang F G, Wang C J, Wang Y M. Digital signature and blind signature based on elliptic curve. J Commun, 2001, 22: 22–28 [张方国, 王常杰, 王育民. 基于椭圆曲线的数字签名与盲签名. 通信学报, 2001, 22: 22–28]
7 Coron J S. On the exact security of full domain hash. In: Advances in Cryptology—CRYPTO 2000. 2000
8 Silverman J H. The Arithmetic of Elliptic Curves. New York: Springer Verlag, 1994
9 Avanzi R, Cohen H, Doche C, et al. Handbook of Elliptic and Hyperelliptic Curve Cryptography. Boca Raton: CRC Press, 2005
10 Du Y S, Zhang F G. Fixed points for elliptic curve discrete logarithms. J Cryptologic Res, 2014, 1: 41–50 [杜育松, 张方国. 椭圆曲线离散对数的不动点. 密码学报, 2014, 1: 41–50]
11 Semaev I A. Summation polynomials and the discrete logarithm problem on elliptic curves. Preprint, 2004. https://eprint.iacr.org/2004/031.pdf
12 Shallue A, van de Woestijne C E. Construction of rational points on elliptic curves over finite fields. In: Algorithmic Number Theory. Berlin: Springer, 2006. 4076: 510–524
13 Ulas M. Rational points on certain hyperelliptic curves over finite fields. Bull Polish Acad Sci Math, 2007, 55: 97–104
14 Icart T. How to hash into elliptic curves. In: Advances in Cryptology—CRYPTO 2009. 2009. 303–316
15 Brier E, Coron J S, Icart T, et al. Efficient indifferentiable hashing into ordinary elliptic curves. In: Advances in Cryptology—CRYPTO 2010. Berlin: Springer, 2010. 237–254
16 Boneh D, Gentry C, Lynn B, et al. Aggregate and verifiably encrypted signatures from bilinear maps. In: Advances in Cryptology—EUROCRYPT 2003. Berlin: Springer, 2003. 416–432
17 Chen Y, Zhao Y. Half-aggregation of Schnorr signatures with tight reductions. In: Computer Security—ESORICS 2022. Cham: Springer, 2022
18 Chalkias K, Garillot F, Kondi Y, et al. Non-interactive half-aggregation of EdDSA and variants of Schnorr signatures. In: Topics in Cryptology—CT-RSA 2021. Cham: Springer, 2021. 577–608
19 Hankerson D, Menezes A, Vanstone S. Guide to Elliptic Curve Cryptography. Berlin: Springer, 2003

Full domain Hash elliptic curve signature

Fangguo ZHANG[1,2]

1. *School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou* 510006*, China*;
2. *Guangdong Province Key Laboratory of Information Security Technology, Guangzhou* 510006*, China*
E-mail: isszhfg@mail.sysu.edu.cn

Abstract The elliptic curve cryptosystem (ECC) remains the most widely used public key cryptosystem. The elliptic curve discrete logarithm problem is its security kernel. We propose a strong fixed point problem of elliptic curve discrete logarithm. Using the strong fixed point assumption, we prove that a full domain hash variant of ECDSA (elliptic curve digital signature algorithm) is secure against existential forgery under the adaptive chosen message attack under the random oracle model. The aggregation properties of signatures make signature schemes play important roles in many scenarios, such as blockchain and cloud storage; therefore, we also discussed the aggregatability of signatures of this variant of ECDSA.

Keywords elliptic curve, digital signature, fixed point, summation polynomial, aggregate signature