

文章编号:1001-9081(2013)01-0163-05

doi:10.3724/SP.J.1087.2013.00163

密码芯片中二元扩域 Eta 双线性对安全算法

柴佳晶^{1*}, 顾海华^{1,2}, 包斯刚¹

(1. 上海华虹集成电路有限责任公司 设计部, 上海 201203; 2. 上海交通大学 计算科学与工程系, 上海 200240)

(* 通信作者电子邮箱 chaijj@shhic.com)

摘要:为了在密码芯片中安全快速地实现二元扩域 Eta 双线性对,提出了基于平方方法的抗功耗攻击实现算法。分别研究了基于平方方法的密钥盲化和明文盲化方案,给出了具体的基于平方方法的抗功耗攻击算法的实现细节。在典型有限域下,基于平方方法的抗功耗攻击算法的实现效率比基于平方根方法提升 10% 以上,并且不需要存储任何预算变量。另外,讨论了将目前用于三元扩域的 Loop Unrolling 方法的思想应用到所提算法后,进一步将运算效率提升了约 3%。效率的提升和存储量的优化使得算法更适用于安全密码芯片。

关键词:Eta 双线性对;二元扩域;抗功耗攻击;密码芯片;效率

中图分类号: TP309.1 文献标志码:A

Security algorithm for Eta bilinear pairing over binary fields in crypto chip

CHAI Jiajing^{1*}, GU Haihua^{1,2}, BAO Sigang¹

(1. Department of Design, Shanghai Huahong Integrated Circuit Company Limited, Shanghai 201203, China;

2. Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China)

Abstract: In order to securely and efficiently realize Eta bilinear pairing over binary fields in crypto chip, a power analysis resistant algorithm was proposed based on square method. The key masking and data masking schemes based on square method were researched respectively, and the implementation details of power analysis resistant algorithm were given based on square method. In typical fields, the implementation efficiency of power analysis resistant algorithm based on square method was increased by 10% or more compared to the algorithm based on square root method, and the proposed algorithm did not need to store any pre-computational variable. Furthermore, the idea of loop unrolling methods in characteristic three was expanded to the proposed algorithm, which further increased the implementation efficiency by about 3%. With the improvement of efficiency and optimization of storage, the proposed algorithm is more suitable for secure crypto chip.

Key words: Eta bilinear pairing; binary field; resistant to power analysis attack; crypto chip; efficiency

0 引言

近年来,双线性对由于其具有双线性性质、非退化性质和可计算性质获得了广泛的研究和应用。双线性对最初在密码学中只是用来攻击椭圆曲线密码系统,但基于双线性对的密码体制以其特有的优点得到了重视和研究,并且在工业界也已逐步应用。国际上许多标准组织也在制定双线性对的标准,例如 ISO/IEC 14888-3, IEEE P1363.3 等。研究者提出了许多基于双线性对的密码方案,例如:基于身份的加密方案^[1], BLS(Boneh, Lynn 和 Shacham)短签名方案^[2], 基于身份的密钥协商方案^[3]等。双线性对构造的密码体制具有独特优点,而这些特点往往是传统的公钥密码体制如 RSA(Rivest, Shamir 和 Adleman)所不具有的,因此可以衍生出许多新的密码应用方向。

在密码芯片中研究双线性对的实现是具有广阔的前景的。然而,相比传统的 RSA 和椭圆曲线密码体制,双线性对在密码芯片中,尤其是在对硬件资源和运算能力有着严格限制的智能卡等安全芯片中的实现还面临着许多现实的问题。其中最为核心的一个问题就是如何在可接受的硬件代价下快速地实现,即如何寻找高效的,适用于硬件特点的实现算法。另一个是工程中十分关键的问题,即如何在确保运算效率的

前提下,找到抵御旁路攻击(Side Channel Attack)的安全实现方法。以往大量的事实表明,即使一个在理论上十分安全的密码算法,如果在硬件实现中不考虑如何抵御旁路攻击,那么其安全性是无法满足工程应用的。旁路攻击,尤其是功耗攻击,几乎破解了所有在数学上安全但实现中不考虑抗攻击措施的密码电路。

双线性对的计算十分复杂,在双线性对友好曲线上能够较为快速地实现双线性对,尤其是在二元扩域 Eta 双线性对的友好曲线上。而二元扩域 Eta 双线性对的实现目前有两种流行的方法,即本文中简称的平方方法和平方根方法。目前国际上许多研究都集中在平方根方法上,并给出了平方根方法抗功耗攻击的优化算法。本文从密码芯片硬件实现的角度进行分析,提出了在平方方法上进行实现将具有更大的优势,给出了基于平方方法的抗功耗攻击优化算法,并且比较了本文所述的方法和平方根方法之间的实现效率和代价。进一步,本文利用目前应用于三元扩域超奇异曲线的 Loop Unrolling 方法的思想对二元扩域 Eta 双线性对运算进行算法优化,给出了算法结果和效率评价。

1 背景

令 F_q 是一个包含 q 个元素的有限域。定义 E 为 F_q 上的一

收稿日期:2012-07-31;修回日期:2012-08-30。 基金项目:2009 年上海科委集成电路设计专项(09706200600)。

作者简介:柴佳晶(1984-),女,上海人,工程师,主要研究方向:安全芯片、密码算法; 顾海华(1981-),男,上海人,工程师,博士,主要研究方向:RSA 密码、椭圆曲线密码的快速安全实现; 包斯刚(1977-),男,上海人,高级工程师,主要研究方向:芯片安全、密码电路。

条椭圆曲线。令整数 r 与 F_q 的特征互素且满足 $r \mid \#E(F_q)$ 。假设 k 是使得 r 整除 $q^k - 1$ 的最小正整数, 称 k 为椭圆曲线 $E(F_q)$ 的嵌入次数。定义 $E(F_{q^k})[r] = \{P \in E(F_{q^k}) \mid [r]P = O\}$ 。 $E(F_{q^k})[r]$ 是 $E(F_{q^k})$ 的一个子群, 其指数为 r 。

约化 Tate 双线性对^[4] 定义如下:

$$\begin{aligned} e_r : E(F_{q^k})[r] \times E(F_{q^k})[r] &\rightarrow \mu_r \\ e_r(P, Q) &\equiv f_{r,P}(D_Q)^{(q^k-1)/r} \end{aligned}$$

其中: $f_{r,P}$ 为椭圆曲线上的有理函数, 满足 $\text{div}(f_{r,P}) = l(P) - l(O)$ 。除了 D_Q 等价于 $(Q) - (O)$; μ_r 为 $F_{q^k}^*$ 中的 r 次单位根群。

Miller^[5] 提出了计算有理函数 $f_{r,P}$ 的有效方法, 但在计算切线时需要进行分母运算。在二元扩域 F_{2^m} 下的超奇异曲线 $E(F_{2^m})$: $y^2 + y = x^3 + x + b$ (下文简称 E_b 曲线) 上, Barreto 等使用了自同态映射 $\psi: E(F_q)[r] \rightarrow E[r]$ ^[6], 能够在 Miller 算法中消除分母运算。

二元扩域 F_{2^m} 下的超奇异曲线 E_b 上的约化 Tate 双线性对^[7] 定义如下:

$$e_t(P, Q) = f_{T,P}(\psi(Q))^{(q^k-1)/l}$$

其中 $P, Q \in E(F_q)[r]$ 。在该曲线上基于 Tate 双线对的变种 Eta 双线性对^[8] 能够使得双线性对的循环次数减少一半。因此 Eta 双线性对成为二元扩域超奇异曲线上计算 Tate 双线对效率最高的方法。二元扩域 F_{2^m} 下, E_b 曲线上两点 $P(x_p, y_p)$ 和 $Q(x_q, y_q)$ 的 Eta 双线性对的定义如下:

$$\eta_T(P, Q) = f_{T,P}(\psi(Q)) =$$

$$l(\psi(Q)) \prod_{i=0}^{(m-1)/2} (g_{[2^i]P}(\psi(Q)))^{\frac{m-1-i}{2}}$$

其中: $T = \mp 2^{\frac{m+1}{2}} - 1$, 函数 $g_P(x, y)$ 为点 P 的切线函数, 函数 $l(x, y)$ 为点 P 的连接线函数。

计算有理函数 $(g_{[2^i]P}(\psi(Q)))$ 的指数 $2^{\frac{m-1-i}{2}}$ 十分不方便, 如果直接做模幂运算效率十分低。目前有两种主流的快速解决方法: 第一种方法将指数运算拆分到函数中, 用多次的扩域模平方来完成指数的运算^[9], 本文将之简称为平方方法; 第二种方法用点半运算代替倍点运算, 因此不再涉及指数运算, 但是需要采用二元扩域 F_{2^m} 中的平方根运算来完成点半运算^[8], 本文将之简称为平方根方法。两种方法相比较, 平方方法每次循环将多涉及 6 次模平方运算^[9], 而平方根方法每次循环将多涉及 2 次平方根运算^[8], 其余的运算量两者类似。从直观的角度来看, 平方根方法中平方根运算的次数远少于平方方法中模平方的运算次数, 并且平方根运算存在优化算法^[10], 能够使其一次运算归约为 0.5 次模乘, 因此平方根方法被广泛地研究和实现。2008 年, Kim 等在平方根方法中加入了随机射影坐标的抗攻击措施, 给出了二元扩域 F_{2^m} 中的超奇异曲线上迄今为止实现效率最高的抗攻击 Eta 双线性对算法^[11]。

相比较而言, 平方方法由于每次循环涉及较多次的模平方运算, 因此很少被学者研究甚至提及, 目前也没有基于该方法的抗攻击实现的论文。然而, 从芯片硬件实现的代价角度来考虑, 平方根方法与平方方法相比却并不具有优势。从以往在智能卡中大量的密码电路实现经验来看, 二元扩域上的

模平方十分快速, 仅涉及用顺序置换运算来实现平方, 如图 1 所示。这样的置换仅涉及电路连线的改变, 不增加任何逻辑电路, 并且不需要增加任何执行时间, 只需要在之后的取模运算中增加简单的控制和调度电路即可。

而二元扩域上的模乘则将涉及用较多数量的异或和移位来实现乘法, 如图 1 所示。假设每个时钟周期计算 $a[i] \times b[j]$ (其中 $a[i]$ 和 $b[j]$ 为 32 比特数据), 那么一共需要 $\lceil a_len/32 \rceil \cdot \lceil b_len/32 \rceil$ 个时钟周期才能完成乘法运算 (a_len 为模乘第一个操作数的比特长度, b_len 为模乘第二个操作数的比特长度)。

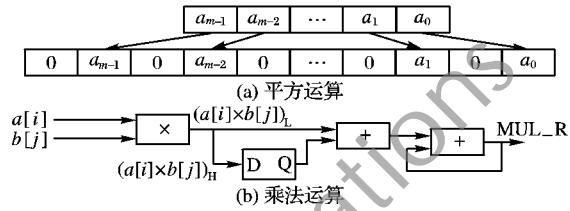


图 1 平方运算和乘法运算的比较

模平方和模乘运算都需要在平方运算和乘法运算之后进行类似的取模运算。美国国家标准与技术研究院 (National Institute of Standards and Technology, NIST) 的标准中都推荐二元扩域上的不可约多项式为三项式或者中间项接近的五项式, 在这种情况下二元扩域上的取模运算能够大大简化。当不可约多项式不同时, 取模运算的具体步骤随项数的不同而略有不同, 但总体的思想类似, 图 2 中的取模运算选用了标准 FIPS186-2 中推荐的有限域 $F_{2^{283}}$ 上的不可约多项式 $f(x) = x^{283} + x^{12} + x^7 + x^5 + 1$ 。从图 2 可以看到, 取模运算只涉及较少量的移位和异或, 并且由于每 32 位只需要一个时钟就能进行取模运算, 因此在有限域 F_{2^m} 上的取模运算只需要大约 $\lceil m/32 \rceil$ 个时钟周期即可实现。

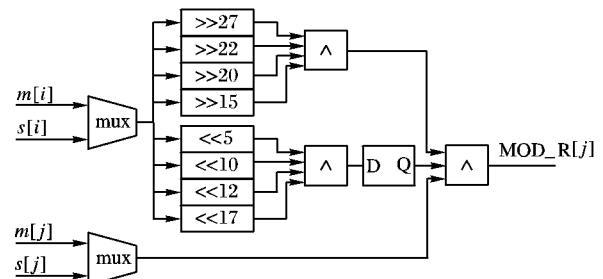


图 2 取模运算

结合两种运算都需要进行的取模运算来考虑, 模平方和模乘的执行时间比一般为 1:9 到 1:16 (取决于基域的选取)。考虑到芯片硬件实现代价以及二元扩域上模平方和模乘速度的巨大差异, 如果芯片中实现了模平方运算, 那么, 平方根方法的硬件执行速度将慢于平方方法的硬件执行速度。此外, 平方根方法中所采用的平方根优化算法^[10]需要预计算变量存储在芯片中才能达到优化的效果, 当芯片支持的有限域较多时, 预计算变量将会占用较多的存储空间。因此从硬件实现的角度来看, 平方方法比平方根方法更适合智能卡等对硬件资源有严格限制的密码芯片。因此, 本文将主要研究基于平方方法的抗攻击实现算法, 通过与文献[11]的结果进行性能对比来阐述本方法的优势, 此外, 还将利用 Loop Unrolling 方法^[12]的思路进一步进行算法优化。

2 二元扩域抗攻击 Eta 双线性对算法

用平方方法计算二元扩域 F_{2^m} 超奇异曲线上 Eta 双线性对的基本形式如算法 1^[9] 所示。

算法 1 Eta 双线性对算法。

二元扩域 F_{2^m} 下曲线 $y^2 + y = x^3 + x + b$, 其中 $b \in F_2$ 。

输入 $P(x_p, y_p), Q(x_q, y_q)$ 。

输出 $\eta_T(P, Q)^{(2^{2m}-1)(2^m-2^{\frac{m+1}{2}}+1)}$ 。

1) $u = x_p + 1$ 。

2) $F = u(u + x_q) + y_p + y_q + b + 1 + (u + x_q)s + t$ 。

3) $C = 1$ 。

4) $x_p = x_p^2 + 1, y_p = y_p^2 + 1, u = y_q + b + 1, v = x_q + 1$,

$\theta = x_p v$ 。

5) i 从 0 到 $\frac{(m-1)}{2}$ 反复执行:

a) $A = y_p + \theta + u + (x_p + v + 1)s + t$;

b) $C = C^2$;

c) $C = C \cdot A$;

d) $x_p = x_p^4, y_p = y_p^4, u = u + v + 1, v = v + 1, \theta = x_p v$ 。

6) $C = C \cdot F$ 。

7) 返回 $C^{(2^{2m}-1)(2^m-2^{\frac{m+1}{2}}+1)}$ 。

功耗攻击就是通过分析 Eta 双线性对运算时的功耗, 利用功耗和密钥之间的相关性获得密钥。按照攻击手段的不同, 可以将其分为简单功耗分析(Simple Power Analysis, SPA) 和差分功耗分析(Differential Power Analysis, DPA)^[13]。

针对双线性对算法的典型攻击场景如下: 在基于身份加密的密码方案中, 在解密阶段需要计算双线性对运算 $e(S_{ID}, U)$, 其中: S_{ID} 是固定的私钥, U 是密文的一部分。攻击者将通过操纵 U , 并用功耗攻击来得到私钥 S_{ID} 。

分析算法 1 描述的 Eta 双线性对算法, 可以发现无论私钥 S_{ID} 和数据 U 为何值, 其算法流程都相同, 整个算法中并没有导致时间差异的分支运算, 只要其硬件实现算法没有因操作数不同而导致时间差异和明显的功耗差异, 那么即使不进行任何改造, 该算法本身也是能够抵抗 SPA 的。

然而, 基于以下的分析, 算法 1 却无法抵御 DPA 攻击。

Eta 双线性对的有理函数为:

$$\begin{aligned} g_{[2^i]P}(\psi(Q)) &= (x_{[2^i]P}^2 + 1)x_q + x_{[2^i]P}x_q + y_{[2^i]P} + y_q + \\ &\quad b + 1 + s(x_{[2^i]P}^2 + x_q + 1) + t \end{aligned}$$

假设点 $P(x_p, y_p)$ 为私钥, 点 $Q(x_q, y_q)$ 为明文(或者密文), 那么函数中模加运算 $(x_{[2^i]P}^2 + x_q)$ 或者模乘运算 $(x_{[2^i]P}^2 + 1)x_q$ 的两个操作数就分别来自私钥和明文(或者密文)。DPA 攻击通过假设私钥的一位或者多位, 根据推算得到的中间数据来进行功耗分组, 运用 DPA 进行分析, 由此得到私钥。

抗 DPA 最有效的方法即为用随机数对密钥或者明文进行盲化。由于所有的中间数据都被随机数盲化, 因此攻击者无法推算任何中间数据, DPA 分析所依赖的中间数据和功耗曲线的相关性分析也就无法进行。假设 $P(x_p, y_p)$ 为密钥, $Q(x_q, y_q)$ 为明文(或者密文); 反之也可以得到类似的算法。

2.1 密钥盲化方案的原理和算法实现

密钥 $P(x_p, y_p)$ 被非零随机数 $r \in F_{2^m}$ 盲化为 $\bar{P}(x_{\bar{p}}, y_{\bar{p}})$,

其中: $x_{\bar{p}} = rx_p, y_{\bar{p}} = ry_p$ 。

密钥盲化后的有理函数为:

$$\begin{aligned} g_{[2^i]P}(\psi(Q)) &= (x_{[2^i]\bar{P}}^2 + r^2)x_q + x_{[2^i]\bar{P}}x_q + y_{[2^i]\bar{P}}^2 + \\ &\quad r^2(y_q + b + 1) + s(x_{[2^i]\bar{P}}^2 + r^2(x_q + 1)) + r^2t \end{aligned}$$

密钥盲化后的有理函数和原先的有理函数存在如下关系:

$$g_{[2^i]P}(\psi(Q)) = r^2 g_{[2^i]P}(\psi(Q))$$

由于 $r \in F_{2^m}$, 而最终模幂中需要计算 $f^{(2^{2m}-1)}$, 因此 r^2 在最终模幂运算后等于 1。因此该密钥盲化方法无需脱盲。对于连接线函数 $l(x, y) = y + \lambda(x + x_p) + y_p + \delta$ 也使用相同的方法进行盲化。基于平方方法的密钥盲化抗攻击 Eta 双线性对算法如算法 2 所示。

算法 2 抗攻击 Eta 双线性对算法(盲化密钥点 P)。

二元扩域 F_{2^m} 下曲线 $y^2 + y = x^3 + x + b$, 其中 $b \in F_2$ 。

输入 $P(x_p, y_p), Q(x_q, y_q)$ 。

输出 $\eta_T(P, Q)^{(2^{2m}-1)(2^m-2^{\frac{m+1}{2}}+1)}$

1) 产生非零随机数 $r \in F_{2^m}, x_{\bar{p}} = rx_p, y_{\bar{p}} = ry_p$ 。

2) $u = x_{\bar{p}} + r$ 。

3) $F = u(u + rx_q) + ry_p + r^2(y_q + b + 1) + (ru + r^2x_q) \cdot s + r^2t$ 。

4) $C = 1$ 。

5) $x_p = x_p^2 + r^2, y_p = y_p^2 + r^2, r = r^2, u = y_q + b + 1, v = x_q + 1, \theta = x_{\bar{p}}v$ 。

6) i 从 0 到 $\frac{(m-1)}{2}$ 反复执行:

a) $A = y_{\bar{p}} + \theta + ru + (x_{\bar{p}} + rv + r)s + rt$;

b) $C = C^2$;

c) $C = C \cdot A$;

d) $x_{\bar{p}} = x_{\bar{p}}^4, y_{\bar{p}} = y_{\bar{p}}^4, r = r^4, u = u + v + 1, v = v + 1, \theta = x_{\bar{p}}v$ 。

7) $C = C \cdot F$ 。

8) 返回 $C^{(2^{2m}-1)(2^m-2^{\frac{m+1}{2}}+1)}$ 。

2.2 明文盲化方案的原理和算法实现

明文 $Q(x_q, y_q)$ 被非零随机数 $r \in F_{2^m}$ 盲化为 $\bar{Q}(x_{\bar{q}}, y_{\bar{q}})$,

其中: $x_{\bar{q}} = rx_q, y_{\bar{q}} = ry_q$ 。

明文盲化后的有理函数:

$$\begin{aligned} g_{[2^i]P}(\psi(\bar{Q})) &= (x_{[2^i]\bar{P}}^2 + 1)x_{\bar{q}} + r \cdot x_{[2^i]\bar{P}}x_{\bar{q}} + r \cdot y_{[2^i]\bar{P}}^2 + \\ &\quad y_{\bar{q}} + r(b + 1) + s(r \cdot x_{[2^i]\bar{P}}^2 + x_{\bar{q}} + r) + rt \end{aligned}$$

密钥盲化后的有理函数和原先的有理函数存在如下关系:

$$g_{[2^i]P}(\psi(\bar{Q})) = r \cdot g_{[2^i]P}(\psi(Q))$$

由于 $r \in F_{2^m}$, 而最终模幂中需要计算 $f^{(2^{2m}-1)}$, 因此 r 在最终模幂运算后等于 1。因此该明文盲化方法无需脱盲。对于连接线函数 $l(x, y) = y + \lambda(x + x_p) + y_p + \delta$ 也使用相同的方法进行盲化。基于平方方法的明文盲化抗攻击 Eta 双线性对算法如算法 3 所示。

算法 3 抗攻击 Eta 双线性对算法(盲化明文点 Q)。

二元扩域 F_{2^m} 下曲线 $y^2 + y = x^3 + x + b$, 其中 $b \in F_2$ 。

输入 $P(x_p, y_p), Q(x_q, y_q)$ 。

输出 $\eta_T(P, Q)^{(2^{2m}-1)(2^m-2^{\frac{m+1}{2}}+1)}$

- 1) 产生非零随机数 $r \in F_{2^m}$, $x_Q = rx_Q, y_Q = ry_Q$ 。
- 2) $u = x_p + 1$ 。
- 3) $F = u(ru + x_Q) + y_Q + r(y_p + b + 1) + (ru + x_Q)s + rt$ 。
- 4) $C = 1$ 。
- 5) $x_p = x_p^2 + 1, y_p = y_p^2 + 1, u = y_Q + r(b + 1), v = x_Q + r, \theta = x_p v$ 。
- 6) i 从 0 到 $\frac{(m-1)}{2}$ 反复执行:
 - a) $A = ry_p + \theta + u + (rx_p + v + r)s + rt$;
 - b) $C = C^2$;
 - c) $C = C \cdot A$;
 - d) $x_p = x_p^4, y_p = y_p^4, u = u + v + r, v = v + r, \theta = x_p v$ 。
- 7) $C = C \cdot F$ 。
- 8) 返回 $C^{(2^{2m}-1)(2^m-2)^{\frac{m+1}{2}}+1}$ 。

无论是经过明文盲化方法或者密钥盲化方法改造之后的 Eta 双线性对算法,由于功耗分组所依赖的中间数据已经被盲化,因此攻击者就无法预计观察点,从而无法进行有效的功耗分组来实施 DPA 攻击。

3 算法效率分析和对比

国际上的众多学者给了几种双线性对的抗攻击方案。Page 等利用双线性对性质盲化运算数据^[14]。Scott 用随机数盲化有理函数运算中的所有中间数据,包括密钥和明文^[15]。Kim 等用随机射影坐标方法对二元扩域 Eta 平方根双线性对算法进行抗攻击改造^[11]。在这些抗攻击方案中,针对二元扩域 Eta 双线性对算法来说,Kim 等在基于平方根方法上实现的抗攻击算法效率最高。在文献[11]中所给出的抗攻击算法的运算量评估中,将平方根运算量等价于平方运算量。但是根据平方根优化算法^[10]的描述,只有当二元扩域 F_{2^m} 上的既约多项式为三项式时,由于平方根运算存在进一步的优化算法,此时其运算时间才和模平方运算等价。而当既约多项式为五项式时,这种等价关系并不成立。NIST 在 FIPS186-2 中推荐的既约多项式中超过一半为五项式。因此从更广泛的适用性来看,平方根的运算量不能简单地等价于平方运算量。因此,该方案经过重新评估后的运算量和本文基于平方方法实现的 Eta 双线性对上的抗攻击算法的运算量比较如表 1 所示(不包括最终模幂运算),其中:M 表示二元扩域 F_{2^m} 下的模乘运算,S 表示二元扩域 F_{2^m} 下的模平方运算,R 表示二元扩域 F_{2^m} 下的平方根运算。

表 1 平方根方法和平方法的运算量评估

Eta 双线性对 计算方法	盲化点 P	盲化点 Q
平方根方法	$(m+1)(7M+S+1.5R)+7M$	$(m+1)(6M+1.5S+R)+5M$
平方方法	$(m+1)(5.5M+5S)+18M+4S$	$(m+1)(5.5M+4S)+15M+2S$

在硬件实现中,平方根运算的时间效率大约为模乘运算的一半,可以等价为 $0.5M$ 。而平方运算的时间效率根据基域不同而略有不同,有限域 $F_{2^{283}}$ 中等价为 $0.11M$,有限域 $F_{2^{367}}$ 中等价为 $0.09M$,有限域 $F_{2^{571}}$ 中等价为 $0.06M$ 。从表 1~2 可以看出,无论是平方方法还是平方根方法,盲化点 Q 都比盲化

点 P 的运算量少。几种典型有限域下平方方法和平方法盲化点 Q 的运算量比如表 2 所示。

表 2 典型有限域下平方根方法和平方法的运算量比较

典型有限域	平方根方法运算量	平方方法运算量	效率提高比例/%
有限域 $F_{2^{283}}$	1 897.86M	1 702.18M	10.31
有限域 $F_{2^{367}}$	2 446.68M	2 171.66M	11.24
有限域 $F_{2^{571}}$	3 774.48M	3 298.40M	12.61

假设在芯片中同时支持上述三个有限域的运算,平方根方法中所采用的平方根优化算法^[10]需要预计算变量存储在芯片中,其存储量为 1 221 比特,而平方方法则不需要存储任何数据。通过分析可知,如果芯片中实现了模平方的电路,从而采用平方方法进行抗攻击的实现,其执行速度相比目前基于平方根方法进行的抗攻击的实现要快。如果再考虑到芯片对于预计算数据的资源开销,平方方法的优势将更为明显。

4 进一步优化

Granger 等提出 Loop Unrolling 方法^[12]对三元扩域超奇异曲线上的双线性对进行优化实现。其基本思想是将双线性对运算循环的两轮归并在一起运算,由于扩域模乘的基域运算量会根据操作数的稀疏形式不同而变化,归并后的运算能够利用稀疏形式来减少基域模乘的运算量。Loop Unrolling 方法在三元扩域中被广泛研究,但由于该方法不能减少二元扩域中未加入抗攻击措施的 Eta 双线性对算法的运算量,因此该方法从未在二元扩域中使用。

当二元扩域 Eta 双线性对算法加入抗攻击措施之后,循环中的扩域模乘操作数的稀疏形式发生变化,因此 Loop Unrolling 方法就能够减少抗攻击 Eta 双线性对算法的运算量。本文提出将 Loop Unrolling 方法引入二元扩域中,进一步优化二元扩域超奇异曲线上 Eta 双线性对抗攻击实现。Loop Unrolling 方法不仅能够加速平方方法的实现,对平方根方法也同样适用。

以平方方法主循环中的运算为例,可以归纳为如下形式:

$$C_i = C_{i-1}^2 \cdot A_i$$

其中平方运算 C_{i-1}^2 的主要运算量为 4 次有限域 F_{2^m} 中的模平方。由于 A_i 的形式稀疏,可以表示成 $A_i = a_2t + a_1s + a_0, a_i \in F_{2^m}$, 该稀疏模乘的主要运算量为 8 次有限域 F_{2^m} 中的模乘。

Loop Unrolling 方法一次性计算主循环中的两轮,可以归纳为如下形式:

$$C_i = (C_{i-2}^2 \cdot A_{i-1})^2 \cdot A_i = C_{i-2}^4 \cdot A_{i-1}^2 \cdot A_i$$

其中:计算 C_{i-2}^4 的主要运算量为 8 次有限域 F_{2^m} 中的模平方,计算 A_{i-1}^2 的主要运算量为 3 次有限域 F_{2^m} 中的模平方。并且 A_{i-1}^2 和 A_{i-1} 具有相同的稀疏形式,因此 $A_{i-1}^2 \cdot A_i$ 的运算为稀疏模乘,其主要运算量为 6 次有限域 F_{2^m} 中的模乘。最后, C_{i-2}^4 和 $A_{i-1}^2 \cdot A_i$ 的乘法运算为有限域 $F_{2^{4m}}$ 中的全模乘,主要运算量为 9 次有限域 F_{2^m} 中的模乘。

通过比较可得,Loop Unrolling 方法使得主循环中的每两次循环减少 1 次 F_{2^m} 中的模乘,增加 3 次 F_{2^m} 中的模平方。虽然直观看来,增加的模平方的次数大于减少的模乘的次数,但是由于二元扩域 F_{2^m} 下的模平方运算十分快速,在芯片中模平方和模乘的执行时间比一般为 1:9 到 1:16(取决于基域的选取)。因此,用 Loop Unrolling 方法改进后的二元扩域超奇异

曲线上的 Eta 双线性对抗攻击算法实现速度更快。

下面以优化算法 3 为例,用 Loop Unrolling 方法改进后的对抗攻击 Eta 双线性对如算法 4 所示。

算法 4 Loop Unrolling 抗攻击 Eta 双线性对算法(盲化明文点 Q)。

二元扩域 F_{2^m} 下曲线 $y^2 + y = x^3 + x + b$, 其中 $b \in F_2$ 。

输入 $P(x_p, y_p), Q(x_q, y_q)$ 。

输出 $\eta_T(P, Q)^{(2^{2m}-1)(2^m-2^{\frac{m}{2}}+1)}$ 。

- 1) 产生非零随机数 $r \in F_{2^m}, x_{\bar{q}} = rx_q, y_{\bar{q}} = ry_q$ 。
- 2) $u = x_p + 1$ 。
- 3) $F = u(ru + x_{\bar{q}}) + y_{\bar{q}} + r(y_p + b + 1) + (ru + x_{\bar{q}})s + rt$ 。
- 4) $C = 1$ 。
- 5) $x_p = x_p^2 + 1, y_p = y_p^2 + 1, u = y_{\bar{q}} + r(b + 1), v = x_{\bar{q}} + r, \theta = x_p v$ 。

6) i 从 0 到 $\lfloor \frac{m-1}{4} \rfloor$ 反复执行:

- a) $A = ry_p + \theta + u + (rx_p + v + r)s + rt$;
- b) $C = C^4, A = A^2$;
- c) $x_p = x_p^4, y_p = y_p^4, u = u + v + r, v = v + r, \theta = x_p v$;
- d) $A' = ry_p + \theta + u + (rx_p + v + r)s + rt$;
- e) $A = A' \cdot A, C = C \cdot A$;
- f) $x_p = x_p^4, y_p = y_p^4, u = u + v + r, v = v + r, \theta = x_p v$ 。
- 7) $C = C \cdot F$ 。

8) 返回 $C^{(2^{2m}-1)(2^m-2^{\frac{m}{2}}+1)}$ 。

算法 4 针对的是 $(m-1)/2$ 为奇数的情况。当 $(m-1)/2$ 为偶数时, 只需要将最后一轮单列出来, 其他轮同样可以用 Loop Unrolling 的方法加速。

第 3 章中的抗攻击 Eta 双线性对算法(算法 3)利用 Loop Unrolling 方法加速的抗攻击 Eta 双线性对算法(算法 4)的运算量在几种典型有限域下的比较如表 3 所示(不包括最终模幂运算, 模平方运算在有限域 $F_{2^{283}}$ 中等价为 0.11M, 有限域 $F_{2^{367}}$ 中等价为 0.09M, 有限域 $F_{2^{571}}$ 中等价为 0.06M)。

表 3 典型有限域下算法 3 和算法 4 的运算量比较

典型有限域	算法 3	算法 4 (Loop Unrolling 方法)	效率提高 比例/%
有限域 $F_{2^{283}}$	1 702.18M	1 657.83M	2.61
有限域 $F_{2^{367}}$	2 171.66M	2 107.68M	2.95
有限域 $F_{2^{571}}$	3 298.40M	3 184.26M	3.46

从表 3 可以看出: Loop Unrolling 方法能够提升二元扩域超奇异曲线上的抗攻击 Eta 双线性对的运算效率; 并且随着应用中安全需求的提高, 运算效率提高的百分比将进一步增大, 优化的效果也将更加明显。

5 结语

本文从密码芯片的硬件实现效率和代价的角度, 研究和分析了二元扩域 F_{2^m} 上基于平方方法实现的抗攻击 Eta 双线性对算法, 该算法相比当前主流的平方根抗攻击方法, 其硬件实现的开销更低(考虑硬件存储参数的开销), 并且算法实现的时间效率更高。在目前密码芯片应用选取的几个典型的有限域中, 基于平方方法实现的抗攻击 Eta 双线性对算法比目前最快的基于平方根算法实现的抗攻击 Eta 双线性对算法的

效率提高了 10% ~ 12%, 并且随着今后应用安全需求的提高, 选取的有限域规模增加, 提高的效果将更为明显。此外, 本文还进一步优化了抗攻击 Eta 双线性对算法, 将 Loop Unrolling 方法引入二元扩域中, 该方法无论对平方方法还是平方根方法都适用, 能够继续提升大约 3% 的运算效率。

参考文献:

- [1] BONEH D, FRANKLIN M. Identity based encryption from the Weil pairing [C]// Advances in Cryptology-CRYPTO 2001: Proceedings of the 21st Annual International Cryptology Conference, LNCS 2139. Berlin: Springer-Verlag, 213 ~ 229.
- [2] BONEH D, LYNN B, SHACHAM H. Short signatures from the Weil pairing [C]// ASIACRYPT'01: Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology. Berlin: Springer-Verlag, 2001: 514 ~ 532.
- [3] SMART N P. An identity based authenticated key agreement protocol based on the Weil pairing [J]. Electronics Letters, 2002, 38 (13): 630 ~ 632.
- [4] 赵昌安, 张方国. 双线性对有效计算研究进展[J]. 软件学报, 2009, 20(11): 3001 ~ 3009.
- [5] MILLER V S. The Weil pairing and its efficient calculation [J]. Journal of Cryptology, 2004, 17(4): 235 ~ 261.
- [6] BARRETO P S, KIM H Y, LYNN B, et al. Efficient algorithms for pairing-based cryptosystems [C]// CRYPTO'02: Proceedings of the 22nd Annual International Cryptology Conference on Advances in Cryptology. Berlin: Springer-Verlag, 2002: 354 ~ 368.
- [7] KWON S. Efficient Tate pairing computation for elliptic curves over binary fields [C]// Proceedings of the 10th Australasian Conference on Information Security and Privacy. Berlin: Springer-Verlag, 2005: 134 ~ 145.
- [8] BARRETO P S, GALBRAITH S D, O'HEIGEARTAIGH C, et al. Efficient pairing computation on supersingular Abelian varieties [J]. Designs, Codes and Cryptography, 2007, 42(3): 239 ~ 271.
- [9] SHU C, KWON S, GAJ K. FPGA accelerated Tate pairing based cryptosystems over binary fields [C]// FPT 2006: IEEE International Conference on Field Programmable Technology. Piscataway: IEEE Press, 2006: 173 ~ 180.
- [10] FONG K, HANKERSON D, LOPEZ J, et al. Field inversion and point halving revisited [J]. IEEE Transactions on Computers, 2004, 53(8): 1047 ~ 1067.
- [11] KIM T H, TAKAGI T, HAN D G, et al. Power analysis attacks and countermeasures on ηT pairing over binary fields [J]. Electronics and Telecommunications Research Institute Journal, 2008, 30(1): 68 ~ 80.
- [12] GRANGER R, PAGE D, STAM M. On small characteristic algebraic tori in pairing based cryptography [J]. LMS Journal of Computation and Mathematics, 2006, 9(3): 64 ~ 85.
- [13] OSWALD E. On side-channel attacks and the application of algorithmic countermeasures [D]. Graz, Austria: Graz University of Technology, 2003.
- [14] PAGE D, VERCAUTEREN F. A fault attack on pairing based cryptography [J]. IEEE Transactions on Computers, 2006, 55 (9): 1075 ~ 1080.
- [15] SCOTT M. Computing the Tate pairing [C]// Proceedings of the 2005 International Conference on Topics in Cryptology. Berlin: Springer-Verlag, 2005: 293 ~ 304.