计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.191200097

抗能量分析的带符号滑动窗口标量乘算法

龚建锋
茂名职业技术学院计算机工程学院　广东 茂名 525011

摘　要　为解决标量乘法运算在施加抗能量分析措施后会降低其运算效率的问题,给出了一种抗能量分析的带符号滑动窗口标量乘算法。该算法首先利用带符号的滑动窗口编码形式对标量重新编码,然后运用预计算、基点掩码与底层域运算的方法来实现标量乘法运算抵抗能量分析攻击,最后采用混合坐标系来完成标量乘运算。性能分析结果表明:该算法能够很好地抵抗简单能量分析、差分能量分析、零值点能量分析及修正能量分析等,并且相比二进制抗能量分析方案与密钥分解抗能量分析方案,所提方案的执行效率也有大幅提升。可见,所提方案能够兼顾安全与效率,较适用于各类资源受限的密码系统中。

关键词:标量乘运算;能量分析;带符号滑动窗口;预计算;底层域运算
中图法分类号　TP309

Resisting Power Analysis Algorithm of Scalar Multiplication Based on Signed Sliding Window

GONG Jian-feng
Department of Computer Engineering, Maoming Polytechnic, Maoming, Guangdong 525011, China

Abstract　In order to resolve the problem that the operating efficiency of scalar multiplication will be reduced after applying the power analysis attacks measures, a resisting power analysis algorithm of scalar multiplication based on signed sliding window is presented. The presented algorithm recodes the scalar with the signed sliding window, and realizes resisting power analysis attacks by combining with the pre-computation, point mask and field operation. Finally, the scalar multiplication is completed in the system of hybrid coordinate. Performance analysis results indicate that the presented algorithm can effectively resist simple power analysis, differential power analysis, zero-value power analysis, and refined power analysis and so on, and the presented scheme also can significantly improved the operating efficiency by comparing with the resisting power analysis scheme of binary expansion and key assignment. It is concluded that the presented scheme can take into account both security and efficiency, and can be applied to kinds of cryptographic systems with limited resource.

Keywords　Scalar multiplication, Power analysis, Signed sliding window, Pre-computation, Field operation

1　引言

椭圆曲线密码(Ellipse Curve Cryptography, ECC)在同等安全条件下所需密钥长度更短,运算速度更快,存储空间更小[1],因此其能较好地用于资源受限但具有高安全需求的密码系统(如智能卡芯片)中。然而,Kocher 提出的能量分析攻击方法对采用椭圆曲线密码算法的智能卡芯片有较大的安全威胁。能量分析攻击方法通过采集智能卡芯片执行密码算法时的能量消耗信息,然后统计分析能量消耗与密钥之间的相关性来获得密钥[2]。能量分析主要分为简单能量分析(Simple Power Analysis, SPA)、差分能量分析(Differential Power Analysis, DPA)、零值点能量分析(Zero-value Power Analysis, ZPA)和修正能量分析(Refined Power Analysis, RPA)[3]。ZPA 攻击和 RPA 攻击是主要用于对 ECC 进行能量分析的方法。标量乘法运算是 ECC 算法实施抵抗能量分析的核心运算,但是施加抵抗能量分析的措施又会降低标量乘法运算的效率,从而造成安全与效率之间的矛盾。目前,解决这一矛盾的主要方法是通过对标量采用不同的编码形式来提升施加抗能量分析措施后的标量乘法运算的效率。国内外研究人员对 ECC 算法的抗能量分析攻击进行了较为丰富的研究。Ma 等[4]给出一种将密钥分解为多个长度相等的短密钥的抗能量分析算法,并在最优组合坐标系下执行标量乘法运算,该方案能够在保证算法安全性的前提下减小标量乘算法的运算量。Wang 等[5]给出基于带符号双基数系统的抗能量分析算法,首先采用带符号的双基数系统对标量编码,再结合基点掩码的方法来实现抗能量分析攻击,这样在保证安全的同时也能减小标量乘算法的运算量。Yang[6]给出基于带符号阶乘展开式的抗能量分析算法,首先利用带符号阶乘展开式对标量编码,再运用折半运算方法减小标量乘算法的运算量,同时利用随机化掩码技术引入随机点实现抗能量分析,这样可有效解决安全与效率矛盾的问题。Li 等[7]给出基于分段 Montgomery 标量乘的抗简单能量分析攻击,所给算法首先利用双标量乘或三标量乘编码方法结合预计算实现标量乘运算,可以有效减小标量乘算法的运算量,然后采用原子化技术来实

基金项目:广东省自然科学面上项目(2016A030313288)
This work was supported by the General program of Natural Science in Guangdong Province(2016A030313288).
通信作者:龚建锋(chunhuaw0801@163.com)

现抵抗简单能量分析，在一定程度上满足了兼顾安全与效率的要求。针对采用椭圆曲线密码的智能卡芯片无法有效兼顾安全和运算效率这一问题，本文提出一种抗能量分析的带符号滑动窗口标量乘算法(Resisting Power Analysis algorithm of scalar multiplication based on Signed Sliding Window，RPAS2W)，该方案首先采用带符号滑动窗口编码形式对标量重新编码，再结合预计算、基点掩码与底层域 2^kR+S，来达到既能保证算法安全又能减小标量乘算法的运算量的目的，能较好地处理 ECC 标量乘法运算中无法同时兼顾安全与运算效率的问题。

2 相关知识

2.1 带符号滑动窗口的标量编码算法

带符号的滑动窗口算法[8]的基本思想为：采用宽度为 $w+1$ 的窗口对标量 k 的二进制编码进行分组，若窗口中的最高比特位为 1，那么说明此窗口的值为负，即该窗口的值需要用补码表示，并且有一个进位。由此可知，标量 k 的带符号滑动窗口编码系数包括 $\{-2^w+1,\cdots,-3,-1,0,1,3,\cdots,2^w-1\}$。文献[9]证明了利用带符号滑动窗口编码能把 $n$bit 二进制序列变换为非零个数最少的编码序列，平均个数为 $\frac{n}{(w+2)}$。标量 k 的带符号滑动窗口编码表示方法如式(1)所示：

$$\begin{aligned} k &= \sum_{j=0}^{t-1} s_j \cdot 2^j \\ &= 2(2\cdots(2(2\cdot s_{t-1}+s_{t-2})+s_{t-3})\cdots+s_1)+s_0 \end{aligned} \tag{1}$$

其中，$s_j \in \{-2^w+1,\cdots,-3,-1,0,1,3,\cdots,2^w+1\}$ 为 k 的编码序列，$j \in \{0,1,\cdots,t-1\}$，t 为编码长度。k 的带符号滑动窗口编码算法如算法 1 所示。

算法 1 标量 k 的带符号滑动窗口编码算法

输入：标量 k

输出：$(s_{t-1},s_{t-2},\cdots,s_j,\cdots,s_1,s_0)$

1. 对于 j 从 0 到 t−1，重复计算：
   1.1 若 $s_j=0$，则计算 j=j+1；
   1.2 若 $s_j=1$，则计算：
      1.2.1 若 $s_{j+w}=0$，计算 $s_j=\sum_{j}^{j+w} k_j \cdot 2^{w-j}$；/* k_j 为二进制编码对应的值 */
      1.2.2 若 $s_{j+w}=1$，则计算：
         1.2.2.1 $s_j=-(\sum_{j}^{j+w} \overline{k_j} \cdot 2^{w-j}+1)$；/* $\overline{k_j}$ 为对二进制编码 k_j 取反 */
         1.2.2.2 $k_{j+w+1}=k_{j+w+1}+1$；
      1.2.3 对于 j 从 j+1 到 j+w+1，有 $s_j=0$；/* 以宽度 w+1 为窗口滑动，计算出 s_j 后，则滑动 w+1 宽度再计算 s_{j+w+1}，中间从 s_{j+1} 到 s_{j+w} 均补为 0 */
      1.2.4 j=j+w+1；
2. 返回 $(s_{t-1},s_{t-2},\cdots,s_j,\cdots,s_1,s_0)$。

2.2 基于带符号滑动窗口的标量乘快速算法

按照式(1)标量 k 的带符号滑动窗口编码表示方法，则有基于带符号滑动窗口的标量乘法运算，如式(2)所示：

$$\begin{aligned} Q=kP &= \left(\sum_{j=0}^{t-1} s_j \cdot 2^j\right) \cdot P = \sum_{j=0}^{t-1} (s_j \cdot P) \cdot 2^j \\ &= \sum_{j=0}^{t-1} P_j \cdot 2^j \end{aligned} \tag{2}$$

其中，$P_j=s_j \cdot P$ 利用预计算得到，即需要构造预计算表 $\{(-2^w+1)P,\cdots,-3P,-P,3P,\cdots,(2^w-1)P\}$，用于在执行标量乘算法主循环运算时调用。基于带符号滑动窗口标量乘算法的步骤如下：1)计算标量 k 的带符号滑动窗口编码；2)生成预计算表 P_j；3)执行主循环运算得到 $Q=kP$。则基于带符号的滑动窗口标量乘算法如算法 2 所示。

算法 2 基于带符号滑动窗口的标量乘算法

输入：k，P

输出：Q=kP

1. 对标量 k 编码得到 $(s_{t-1},s_{t-2},\cdots,s_j,\cdots,s_1,s_0)$；/* 根据算法 1 求得 */
2. 生成预计算表 P_j；/* 预计算表 P_j 的生成算法详见算法 3 */
3. 假定 Q=O；
4. 对于 j 从 t−1 到 0，重复计算：
   4.1 Q=2Q；/* 根据式(2)的标量乘运算可知，无论系数 s_j 为何值，均需先执行倍点运算 */
   4.2 若 $s_j>0$，则计算 $Q=Q+P_j$；/* 当系数 s_j 为正时，直接与 P_j 进行点加运算 */
   4.3 若 $s_j<0$，则计算 $Q=Q+|P_j|$；/* 当系数 s_j 为负时，需先求得 $|P_j|=-P_j$，再与 $|P_j|$ 进行点加运算 */
5. 返回 Q。

算法 3 给出了 P_j 的构造方法。

算法 3 P_j 的构造算法

输入：w，P /* 其中 w+1 即为窗口宽度 */

输出：P_j /* 其中 $P_j=s_j \cdot P$ */

1. 假定 $P_1=P$，$|P_1|=-P_1$；
2. 假定 $P_2=P+P=2P$，$|P_2|=-P_2$；
3. 对于 j 从 1 到 $2^{w-1}-1$，重复计算：
   3.1 $P_{2j+1}=P_{2j-1}+P_2$；/* 由于标量编码系数 $s_j \in \{-2^w+1,\cdots,-3,-1,0,1,3,\cdots,2^w+1\}$，因此预计算表中的各分基点需要通过 $P_{2j+1}=P_{2j-1}+P_2$ 计算求得 */
   3.2 $|P_{2j+1}|=-P_{2j+1}$；/* 当标量编码系数 s_j 为负时，通过该转换可直接进行点加运算 */
4. 返回 P_j。

3 RPAS2W 算法设计

RPAS2W 算法的流程是：首先，根据带符号滑动窗口编码算法(算法 1)对 k 编码，可得到非 0 比特位数量最少的编码序列，即 $k=\sum_{j=0}^{t-1} s_j \cdot 2^j$，其中 $s_j \in \{-2^w+1,\cdots,-3,-1,0,1,3,\cdots,2^w+1\}$；其次，通过引入随机点 R 来随机盲化基点，同时把随机点应用到预计算中来构造新的预计算表，即令 $P_1=P-R$，根据算法 3 生成采用随机点掩码后的预计算表 Z_j；最后，通过底层域计算 2^kR+S[10]实现标量乘法运算，这样既能提高标量乘法运算的效率，还能将算法中的点加运算与倍点运算实现归一化，这样可以使标量乘算法在执行过程中不会出现能量消耗差异。在主循环运算中令 $Q=R$，可消除在预计算中引入随机点的影响，恢复真实返回值。抗能量分析的带符号滑动窗口标量乘算法(RPAS2W 算法)如算法 4 所示。

算法 4 抗能量分析的带符号滑动窗口标量乘算法

输入:k,P

输出:Q=kP

1. 对标量 k 编码得到(s_{t-1},s_{t-2},…,s_j,…,s_1,s_0);/* 根据算法 1 求得 */
2. 引入随机点 R=random();/* 引入的一个随机点,主要用于掩盖基点 */
3. 生成预计算表 Z_j;/* 预计算表 Z_j 中引入了随机点,其生成算法详见算法 5 */
4. 假定 Q=R;/* R 为引入的随机点 */
5. 对于 j 从 t−1 到 0,重复计算:
   5.1 若 $s_j>0$,则计算 $Q=2^jQ+Z_j$;/* 当系数 s_j 为正时,直接执行底层域 2^kR+S 运算 */
   5.2 若 $s_j<0$,则计算 $Q=2^jQ+|Z_j|$;/* 当系数 s_j 为负时,将 $-Z_j$ 转换为正的 $|Z_j|$ 后再执行底层域 2^kR+S 运算 */
6. 返回 Q。

算法 5 给出了 Z_j 的构造方法。

算法 5 Z_j 构造算法

输入:w,P /* 此时 w+1 仍为窗口宽度 */

输出:Z_j/* 除引入了随机点 R,预计算表 Z_j 的计算方法与预计算表 P_j 相同,预计算表的分基点个数相同,且分基点的下标也均对应相同 */

1. 引入随机点 R=random();
2. 假定 $Z_1=P-R$,$|Z_1|=-Z_1$;/* 在 Z_1 中引入随机点 R 后,在后续的循环运算中计算 Z_j 时,可实现预计算表中所有求得的分基点均被掩盖 */
3. 假定 $Z_2=(P-R)+(P-R)=2Z_1$,$|Z_2|=-Z_2$;
4. 对于 j 从 1 到 $2^{w-1}-1$,重复计算:
   4.1 $Z_{2j+1}=Z_{2j-1}+Z_2$;/* 由于标量编码系数 $s_j\in\{-2^w+1,\cdots,-3,-1,0,1,3,\cdots,2^w+1\}$,因此预计算表中的各分基点需要通过 $Z_{2j+1}=Z_{2j-1}+Z_2$ 计算求得 */
   4.2 $|Z_{2j+1}|=-Z_{2j+1}$;/* 当标量编码系数 s_j 为负时,通过该转换可直接进行点加运算 */
5. 返回 Z_j。/* 得到引入了随机点的预计算表 */

4 性能分析

4.1 安全性分析

下文将分别分析所提 RPAS2W 算法抵抗 SPA,DPA,ZPA 与 RPA 4 种能量分析的能力,详细分析过程如下。

(1)抵抗简单能量分析攻击(SPA)的性能分析。传统标量乘算法是在运算过程中,当二进制编码系数为 1 时执行点加操作,当二进制编码系数为 0 时不执行点加操作,因此产生了能耗差异。SPA 攻击通过不同的能耗来猜测标量乘运算是否有执行点加操作,如果执行点加操作,则可知二进制编码系数为 1,否则二进制编码系数为 0,由此可得到完整的二进制编码信息,即可获得密钥信息。RPAS2W 算法(算法 4)在步骤 5 中,由于采用了底层域直接计算 2^kR+S 方法,对主循环运算中的点加运算与倍点运算进行了归一化,使得无论编码系数为何值,每次均会执行一次 2^kR+S 操作,即标量乘运算每次操作的能耗均是相同的,其能耗曲线将不会出现明显的差异,因此采用 SPA 攻击无法获得密钥信息,即 RPAS2W 算法能抵抗简单能量分析。

(2)抵抗差分能量分析攻击(DPA)的性能分析。攻击者实施 DPA 攻击主要通过猜测不同的密钥位或是通过输入不同的基点来实施大量的标量乘运算,根据所获得的大量能耗曲线与实际密钥的标量乘运算能耗曲线进行差分计算,从而得到一条差分能耗曲线,如果所得差分能耗曲线没有出现明显尖峰,则猜测的密钥位是正确的,反之,如果所得差分能耗曲线出现明显尖峰,则说明猜测的密钥是错误的。但是在所给 RPAS2W 中步骤 5 的主循环运算中,由于无论编码系数为何值,RPAS2W 算法主循环运算均只执行底层域 2^kR+S 运算,攻击者无论猜测何种密钥,每次执行主循环运算时的能耗曲线均是相同的,因此攻击者获得的差分能耗曲线不会出现明显起伏尖峰,即使输入不同的基点进行标量乘运算,最终所获得的差分能耗曲线也同样是平滑无尖峰的,因此攻击者无法由已获得的能量消耗曲线进行差分统计分析,进而无法获得与密钥相关的有效信息,即 RPAS2W 算法能抵抗差分能量分析。

(3)抵抗零值点能量分析攻击(ZPA)的性能分析。在传统二进制标量乘算法中,当执行点加操作时,在寄存器中存储该值,同时可知此时的密钥位为 1;当不执行点加操作时,在寄存器中不需存储数值,此时可知密钥位为 0,ZPA 即是通过这种方式实施攻击的。但在 RPAS2W 算法的步骤 5 中,由于无论编码系数 s_j 取何值,均需执行一次底层域 2^kR+S 运算,每次循环运算都需要在存储器中存入中间值,即 RPAS2W 算法掩盖了密钥与中间值之间的关系,使得攻击者无法根据寄存器存储数据来猜测密钥,因此 RPAS2W 算法可以抵抗零值点能量分析攻击。

(4)抵抗修正能量分析攻击(RPA)的性能分析。RPA 攻击主要通过将椭圆曲线上的点投影到坐标系来得到特殊点$(x,0)$或$(0,y)$,以此获取此时的密钥信息,然后根据该密钥位的能耗信息与其他密钥位的信息进行差分,如果其他密钥位的能耗信息与该密钥位的能耗信息相同,则表明两个密钥位相同,反之,则表明两个密钥位不同,由此可获取标量乘算法的密钥信息。但是在 RPAS2W 算法的步骤 2 和步骤 4 中,由于预计算 Z_j 和 $Q=R$ 引入了随机点 R,而攻击者无法获知 R,使得攻击者投影到坐标系的特殊点不是正确的,获取的密钥也不是真实密钥,即隐藏了中间运算结果与能耗之间的关系,其他密钥位的能耗信息与该密钥位的能耗信息进行差分所得到的密钥位也不是真实密钥位,因此 RPAS2W 算法能抵抗修正能量分析攻击。

4.2 效率分析

RPAS2W 算法(算法 4)的步骤 1 中,采用算法 1 对 k 进行重新编码所需的运算量较小,相比标量乘算法的运算量其能够忽略。步骤 3 中,生成 T_j 需要执行 1 次倍点操作与 2^{w-1} 次点加操作,其运算量为 $D_A+2^{w-1}\cdot A_A$(其中,下标 A 在仿射坐标下计算)。步骤 5 执行 $n/(w+2)$ 次底层域运算 2^kR+S,其运算量为 $n/(w+2)\cdot DDA_{JA}$(其中,下标 JA 在仿射+Jacobi 混合坐标系下计算)。A 为点加操作,D 为倍点操作,DDA 为底层域 2^kR+S。3 种抗能量分析算法(二进制抗能量分析算法、密钥分解抗能量分析算法、所提 RPAS2W 算法)的运算量比较结果如表 1 所列。

表 1 所提 RPAS2W 算法与传统抗能量分析方案的运算量比较

Table 1 Calculation comparison of RPAS2W algorithm and traditional resisting power analysis scheme

算法	预计算量	主运算量	总运算量
二进制法	A_J	nA_J+nD_J	$nA_J+(n+1)D_J$
密钥分解法	$(2m-1)A_A+(m-1)(n/m)D_J+(m-2)t_{J\to A}$	$(n/m+1)A_{JA}+(n/m)D_J$	$(2m-1)A_A+(n/m+1)A_{JA}+nD_J+(m-2)t_{J\to A}$
本文算法 4	$A_A+2^{w-1}D_A$	$[n/(w+2)]DDA_{JA}$	$A_A+2^{w-1}D_A+[n/(w+2)]DDA_{JA}$

注：文献[3]给出的基于密钥分解抗能量分析标量乘算法中，$t_{J\to A}$ 的计算量为 $44M$，m 表示密钥分段的个数，且 $m=4$

在不同的坐标系下，椭圆曲线上点加运算和倍点运算的运算量不相同。文献[11]给出了椭圆曲线上点加运算和倍点运算在不同坐标系下的运算量，如表 2 所列。表 2 中，A 表示仿射坐标系，J 表示 Jacobi 坐标系，P 表示标准映射坐标系，C 表示 Chudonvsky 坐标系，$A+A\to A$ 表示在仿射坐标系下的点加运算，$A+J\to J$ 表示在仿射＋Jacobi 混合坐标系下的点加运算，$2A\to A$ 表示在仿射坐标系下的倍点运算，其余符号的含义以此类推。

表 2 不同坐标系下椭圆曲线上点加运算和倍点运算的运算量

Table 2 Operations of point addition and multiple point on elliptic curves in different coordinate systems

点加运算		倍点运算	
不同坐标系运算	运算量	不同坐标系运算	运算量
$A+A\to A$	S+2M+I	$2A\to A$	2S+2M+I
$J+J\to J$	4S+12M	$2J\to J$	6S+4M
$P+P\to P$	4S+12M	$2P\to P$	5S+7M
$C+C\to C$	3S+11M	$2C\to C$	6S+5M
$A+J\to J$	3S+8M	—	—
$A+C\to C$	3S+8M	—	—
$C+J\to J$	3S+11M	—	—

由表 2 可知，仿射坐标系下，A_A 的运算量为 S+2M+I，D_A 的运算量为 2S+2M+I；Jacobi 坐标系下，A_J 的运算量为 4S+12M，D_J 的运算量为 6S+4M；仿射＋Jacobi 混合坐标系下，A_{JA} 的运算量为 3S+8M，DDA_{JA} 的运算量为 $(4w+3)$S+$(4w+9)$M[12]。M 表示模乘运算，S 表示平方运算，I 表示模拟运算。文献[13]指出，根据坐标系的选择和密钥长度的不同，模拟运算 I 与模乘运算 M 的比例关系为 9～30，即 $\frac{\mathrm{I}}{\mathrm{M}}\in(9,30)$，但在实际运用中通常选取 I≈20 M 与 S≈M。则可得 RPAS2W 算法的总运算量 C_T 如式(3)所示：

$$\begin{aligned}C_T&=D_A+2^{w-1}A_A+[n/(w+2)]DDA_{JA}\\&=(2\mathrm{S}+2\mathrm{M}+\mathrm{I})+2^{w-1}(\mathrm{S}+2\mathrm{M}+\mathrm{I})+[\mathrm{n}/(w+2)]\\&\quad[(4w+3)\mathrm{S}+(4w+9)\mathrm{M}]\\&=24\mathrm{M}+2^{w-1}\cdot 23\mathrm{M}+[n/(w+2)](8w+12)\mathrm{M}\end{aligned}\quad(3)$$

其中，$w+1$ 为窗口宽度，当窗口宽度取 4 时的运算量最小，则有 $w=3$。因此，RPAS2W 算法的总运算量 $C_T=(208+7.2n)$M。同理，可计算出二进制抗能量分析标量乘算法的总运算量为 $C_B=(16+26n)$M，基于密钥分解抗能量分析标量乘算法的总运算量为 $C_M=(260+12.75n)$M。由于密钥长度一般较大，因此总运算量的大小与密钥长度 n 有关，则可知 $C_T<C_M<C_B$，RPAS2W 算法所需总运算量最小，且随着密钥长度的增加，RPAS2W 算法的总运算量增加幅度较小。当密钥长度 $n=160$，$n=192$，$n=224$，$n=256$ 和 $n=280$ 时，RPAS2W 方案与二进制抗能量分析标量乘方案和密钥分解抗能量分析标量乘方案的运算量对比结果如图 1 所示。

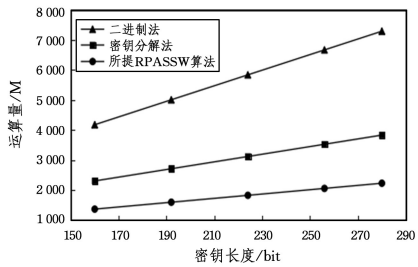


图 1 所提 RPAS2W 算法与传统抗能量分析方案在不同密钥长度下的运算量对比

Fig. 1 Calculation comparison of RPAS2W algorithm and traditional resisting power analysis scheme under different key length

一般情况下，在 ECC 算法中采用 512bit 的密钥可以保证安全，即 $n=512$。由表 2 可知，二进制抗能量分析算法在密钥长度为 512bit 时，其预计算量为 16M，主运算量为 13312 M，总运算量为 13328 M；密钥分解抗能量分析算法在密钥长度为 512bit 时，其预计算量为 4273M，主运算量为2699 M，总运算量为 6972 M；RPAS2W 算法在密钥长度为 512bit 时，其预计算量为 116 M，主运算量为 3686.4 M，总运算量为3802.4 M。由此可知，RPAS2W 算法的总运算效率与二进制抗能量分析算法相比提高了 71.5%，与密钥分解抗能量分析算法相比提高了 45.5%。另外，RPAS2W 算法的预计算运算效率与密钥分解抗能量分析算法相比提高了 97.3%。由此可知，RPAS2W 算法所需存储空间相对更小，说明将 RPAS2W 算法应用在资源有限的密码系统中将会更加高效。综上所述，RPAS2W 算法不仅能够抵抗简单能力分析、差分能量分析、零值点能量分析与修正能量分析等，而且标量乘法运算的效率也有了较大提高，所需存储空间也较小，这表明 RPAS2W 算法可同时有效兼顾安全与效率。

结束语 能量分析因其具有实现简单、成功率高等优点，而被广泛应用于 ECC 中实施攻击，但在 ECC 算法中施加抗能量分析措施后将会使标量乘法运算效率降低，导致 ECC 算法存在安全与效率矛盾的问题。标量乘法运算是 ECC 算法中最核心的运算，文中采用带符号滑动窗口编码形式对标量进行编码，可以大幅减少编码中的非零位个数，并且采用预计算、基点掩码与底层域直接计算的方法，不仅能抵抗多种能量分析，而且能大幅提升运算效率。与传统二进制抗能量分析算法以及密钥分解抗能量分析算法相比，所提算法能够在同等安全条件下进一步提升 ECC 算法标量乘法运算的效率，因此所提算法可实际应用于同时对安全性和运算效率要求较高的各个领域。下一步将通过优化带符号滑动窗口标量乘算法来进一步提高所提算法的运算效率。

参 考 文 献

[1] GUO B,SUN Z T,WANG Y,et al. Resisting power analysis attacks algorithm of scalar multiplication based on factorial expansions form [J]. Bulletin of Science and Technology, 2016, 32(6):149-153.

[2] WU K K,LI H Y,YAN L J. Homogeneous mapping model of ECC for preventing differential power analysis[J]. Computer Engineering,2017,43(10):115-119.

[3] LIANG F,SHEN J N. Resisting power analysis attacks scheme for ellipse curve cryptography based on odd-only Comb method [J]. Computer Applications and Software, 2016, 33(3): 288-290.

[4] MA B,BAO S G,DAI X Y. Efficiency improvement of ECC resisting power attack scheme in smart card[J]. Computer Engineering,2010,36(16):113-115.

[5] WANG Z Y,ZHAO J G. Resisting power analysis attack scheme based on signed double-based number system[J]. Journal of Computer Applications,2011,31(11):2973-2974.

[6] YANG B. Secure and efficient scalar multiplication algorithm with power analysis attack resistance[J]. Control Engineering of China,2017,24(12):2462-2465.

[7] LI Y,WANG J L,ZENG X W,et al. A segmented Montgomery scalar multiplication algorithm with resistance to simple power analysis SPA attacks[J]. Computer Engineering and Science, 2017,30(1):92-101.

[8] SHI L,XU M. DWNAF:a dynamic window NAF scalar multiplication with threshold[J]. Computer Science, 2017, 44(10): 159-164.

[9] PHILLIPS B J,BURGESS N. Implementing 1024-bits RSA exponentiation on a 32-bits processor core[C]// Proceeding of the Application Specific-Systems,Architecture and Processor. 2000: 127-137.

[10] WEI G H,WANG Y,ZHANG H G. ECC point multiplication lightweight improvement for RFID applications over GF(2^m) [J]. Computer Engineering and Science,2017,39(1):81-85.

[11] WANG Y X,ZHANG C R,ZHANG B H,et al. Efficient scalar multiplication of ECC based on composite operations over prime fields[J]. Application Research of Computers, 2013, 30(11): 3365-3387.

[12] LIU G Z,QI H X. Efficient NAF scalar multiplication algorithm with low storage [J]. Science Technology and Engineering, 2013,13(19):5683-5686.

[13] BARUA R,PANDEY S K,PANKAJ R. Efficient window-based scalar multiplication on elliptic curves using double-base number system [J]. Lecture Notes in Computer Science,2007,4859(12): 351-360.

GONG Jian-feng, postgraduate, lecturer. His main research interests include computer network technology and information security.

(上接第 513 页)

[11] GOODFELLOW I,SHLENS J,SZEGEDY C,et al. Explaining and Harnessing Adversarial Examples[C]// International Conference on Learning Representations. 2015.

[12] KURAKIN A,GOODFELLOW I,BENGIO S,et al. Adversarial examples in the physical world[C]// International Conference on Learning Representations. 2017.

[13] MOOSAVI-DEZFOOLI S M,FAWZI A,FROSSARD P. Deepfool:A Simple and Accurate Method to Fool Deep Neural Networks[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016:2574-2582.

[14] CARLINI N,WAGNER D. Towards Evaluating the Robustness of Neural Networks[C]// IEEE Symposium on Security and Privacy. 2017:39-57.

[15] MENG D,CHEN H. MagNet:A Two-Pronged Defense against Adversarial Examples[C]// Computer and Communications Security. 2017:135-147.

[16] GU S,RIGAZIO L. Towards Deep Neural Network Architectures Robust to Adversarial Examples[J]. arXiv: Learning, 2014.

[17] PAPERNOT N,MCDANIEL P,WU X,et al. Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks[C]// IEEE Symposium on Security and Privacy. 2016: 582-597.

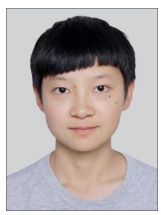
[18] XU W,EVANS D,QI Y,et al. Feature Squeezing Mitigates and Detects Carlini/Wagner Adversarial Examples[J]. arXiv:Cryptography and Security,2017.

[19] XU W,EVANS D,QI Y,et al. Feature Squeezing:Detecting Adversarial Examples in Deep Neural Networks[C]// Network and Distributed System Security Symposium. 2018.

[20] WONG E,RICE L,KOLTER J Z,et al. Fast is Better than Free:Revisiting Adversarial Training[J]. arXiv preprint arXiv: 2001.03994,2020.

[21] XIAO C,ZHONG P,ZHENG C,et al. Enhancing Adversarial Defense by k-Winners-Take-All[J]. arXiv preprint arXiv:1905.10510,2019.

[22] ZANTEDESCHI V,NICOLAE M,RAWAT A,et al. Efficient Defenses Against Adversarial Attacks[J]. arXiv: Learning, 2017.



WANG Dan-ni, born in 1995, postgraduate. Her main research interest includes information security of artificial intelligence.

CHEN Wei, born in 1978, Ph.D, associate professor. His main research interest includes network security.