针对滑动窗口算法的椭圆曲线密码故障分析

张金中[1]，寇应展[1]，王韬[1]，郭世泽[2]，赵新杰[1]
(1. 军械工程学院 计算机工程系，河北 石家庄 050003；2. 北方电子设备研究所，北京 100083)

摘 要：基于符号变换故障攻击原理，针对采用滑动窗口算法实现点乘运算的椭圆曲线密码，当故障位于倍点运算时，给出一种能够解决“零块失效”问题的改进故障分析方法，实验结果表明 15 次故障注入即可恢复 192bit 完整密钥；当故障位于加法运算时，提出一种新的故障分析方法，实验结果表明 1 次故障注入可将密钥搜索空间降低 2^7~2^{15}。该方法对其他使用滑动窗口算法的密码算法故障攻击具有借鉴意义。

关键词：公钥密码；椭圆曲线密码；故障攻击；点乘运算；滑动窗口算法；零块失效
中图分类号：TP393.08 **文献标识码**：A **文章编号**：1000-436X(2012)01-0071-08

Fault analysis on elliptic curve cryptosystems with sliding window method

ZHANG Jin-zhong[1], KOU Ying-zhan[1], WANG Tao[1], GUO Shi-ze[2], ZHAO Xin-jie[1]
(1. Department of Computer Engineering, Ordnance Engineering College, Shijiazhuang 050003, China;
2. The Institute of North Electronic Equipment, Beijin na)

Abstract: Based on the principle of sign change fault attacks, an improved fault analysis method was presented against elliptic curve cryptosystems with sliding window method for point multiplication. When faults on double instruction it could effectively solve the “zero block failure” problem. Experiment results demonstrate that 15 times fault injections are enough to recover full 192bit key. When faults on add instruction, a new fault analysis method was proposed. Experiment results demonstrate one fault injection can reduce searching space 2^7~2^{15}. The method presented here can provide some ideas for fault attack on other cryptosystems using sliding window method.

Key words: public key cryptography; elliptic curve cryptosystems; fault attacks; point multiplication; sliding window method; zero block failure

1 引言

现代密码算法可用于保证数据传送的保密性、完整性和真实性，因而被广泛应用于政治、经济和军事活动等领域。但是密码算法的实际安全性不等同于设计安全性，密码算法执行所依附的物理平台会泄露出时间、电磁、功耗、声音、故障等与密钥密切相关的旁路信息，利用这些旁路信息进行密钥破解的方法称为旁路攻击[1]。其中，故障攻击主要利用密码算法执行过程中的故障信息，结合算法结构破解密钥。从故障攻击提出至今，研究者先后对 RSA[2]，AES[3]、SMS4[4]、MIBS[5]、RC4[6]等密码进行了故障攻击。显然，不论公钥密码、分组密码还是流密码，都面临着

收稿日期：2011-01-17；**修回日期**：2011-07-02
基金项目：国家自然科学基金资助项目(60772082)；河北省自然科学基金资助项目(08M010)
Foundation Items: The National Natural Science Foundation of China (60772082); The Natural Science Foundation of Hebei Province (08M010)

故障攻击的严重威胁。

椭圆曲线密码（ECC，elliptic curve cryptosystem）是近年来密码学领域研究的热点之一，有密钥短、安全性高、计算速度快的特点，已被广泛应用于移动通信、电子商务等领域，也是卫星网络、物联网等新型网络的首选，对其安全性进行研究显得尤为重要。自1999年Coron[7]提出针对ECC的简单功耗分析（SPA, simple power analysis）和差分功耗分析（DPA, differential power analysis）开始，ECC的功耗攻击及相关的防御措施已经逐步走向成熟，但ECC故障攻击相关研究相对较少。

1.1 相关工作

目前针对ECC故障攻击方法主要有3类。

1) ECC差分故障攻击[8]。通过注入故障改变基点、基域或曲线参数，使得破解密钥变为解决非安全椭圆曲线上的离散对数问题。研究者先后攻击了IEEE 802.15制定的加密方案（ECDH、ECIES和ECMQV）[9]和蒙哥马利算法实现的ECC[10]。

2) 符号变换故障攻击[11]。通过注入故障改变乘运算中间变量的符号，Blömer成功攻击了采用二进制非相邻表示（NAF_2）算法实现的ECC，但是密钥恢复算法中仍存在“零块失效”（密钥中连续大量的零位所导致的密钥恢复算法失效）问题。

3) 基于“漏操作”(skip instruction）的故障攻击[12]。使椭圆曲线签名算法在点乘运算执行过程中少执行几次循环，有研究者通过分析故障签名获取部分密钥，最终结合攻击获取完整密钥。

分析表明，1)类、3)类攻击方法的故障信息偏离了原有安全椭圆曲线，可通过检测输入、中间变量和输出是否在原安全椭圆曲线上来进行防御[13]。由于符号变换故障并不使故障点偏离原安全椭圆曲线，上述措施对2)类攻击是失效的。本文选取广泛应用的基于滑动窗口算法的ECC为研究对象，在点乘运算过程中引入符号变换故障，结合算法分析获取密钥。

1.2 本文工作

通过分析滑动窗口算法查表特性，分别基于倍点和加法运算的符号变换故障模型，给出了这2种故障模型下的密钥恢复过程，主要创新点如下。

1) 将符号变换故障攻击思想引入至滑动窗口算法实现的ECC，在故障位置未知情况下，讨论了倍点运算故障下的密钥恢复过程，给出了能有效消除文献[11]中“零块失效”的密钥恢复算法。仿真实验表明：10~20个故障可在30min左右恢复完整NIST-192bit密钥。因为这种故障分析方法主要针对倍点运算，所以对基于平方乘算法、固定窗口算法实现的ECC同样具有威胁。

2) 基于滑动窗口算法查表特性，提出了针对查表操作的故障分析方法。仿真实验表明：当故障精确注入到每次查表操作时，1min即可恢复完整密钥。当故障仅注入到部分查表操作时，可恢复部分密钥，1个故障可降低2^7~2^{15}的密钥空间。

2 针对滑动窗口算法的故障攻击

2.1 符号说明

GF(p), E：有p个元素的有限域，椭圆曲线为E。

P_1, P_2：椭圆曲线上的点为P_1, P_2。

P, k, Q, R：基点为P，私钥为k，正确公钥为Q，故障公钥为R。

k_i：私钥NAF_2表示的第i位为k_i。

w, t：最大窗口宽度为w，算法执行中窗口的实际宽度为t。

t_i：第i轮循环中窗口宽度为t_i。

u, P_u：查表索引为u，查表得到的预计算值为P_u。

Q_i：计算公钥的中间变量为Q_i。

2.2 椭圆曲线密码

椭圆曲线密码是一种基于椭圆曲线离散对数问题的公钥密码。定义在GF(p)上的椭圆曲线E为：$y^2=x^3+ax+b$，其上的点可构成一个有限可交换群。令$P_1=(x_1, y_1)\in E$，则$-P_1=(x_1,-y_1)\in E$，若$P_2=(x_2, y_2)\in E$，则$P_1+P_2=(x_3,y_3)$，$x_3=(?^2-x_1-x_2)\bmod p$, $y_3=(?(x_1-x_3)-y_1)\bmod p$，其中：

$$l=\begin{cases}((y_2-y_1)/(x_2-x_1))\bmod p, & P_1\neq P_2\\ ((3x_1^2+a)/2y_1)\bmod p, & P_1=P_2\end{cases} \tag{1}$$

前者$P_1?P_2$的情形称为点加运算；后者$P_1=P_2$的情形称为点倍运算。对于给定椭圆曲线E,曲线上一点P以及正整数k，点kP计算称为椭圆曲线上的点乘（标量乘），ECC主要根据点乘运算生成公钥$Q=kP$。目前点乘运算的实现方式主要有平方乘算法、蒙哥马利算法、固定窗口算法和滑动窗口算

法等[14]。滑动窗口算法用二进制非相邻表示型表示私钥 $k=\sum_{i=0}^{n-1}k_i 2^i$，其中，$k_i\in\{-1,0,1\}$，该方法可有效降低密钥汉明重，提高计算效率，并能抵御计时[15]和功耗旁路攻击[16]。本文主要针对滑动窗口算法展开故障分析研究。

算法 1　滑动窗口算法

输入：最大窗口 w，私钥 $NAF_2(k)$，基点 P；

输出：公钥 $Q=kP$；

1) 对于 $i\in\{1,3,\cdots,2(2^w-(-1)^w/3-1)\}$，计算 $P_i=iP$；

2) Q 为无穷远点，$i=l-1$；

3) while ($i\geq 0$) do；

① 若 $k_i=0$, 则 $t=1$, $u=0$；

② 否则，寻找满足 $t\leq w$ 且 $u=(k_i,\cdots,k_{i-t+1})$ 为奇数时 t 的最大值；

③ $Q=2^tQ$；

④ 若 $u>0$, 则 $Q=Q+P_u$；

若 $u<0$, 则 $Q=Q-P_{-u}$；

⑤ $i=i-t$；

4) 返回 Q。

滑动窗口算法引入预计算表以提高密码执行效率，但由于每次查表索引即为窗口中密钥绝对值，攻击者如果能够在查表过程中注入故障，密钥信息极有可能随之泄露。此外，算法引入了变量 t 和 u 来控制滑动窗口大小和滑动距离，使得不同密钥对应窗口大小和滑动次数不同，增加了故障分析难度。下文将通过建立故障攻击模型，进一步讨论不同故障下的密钥恢复过程。

2.3　针对滑动窗口算法的故障模型

故障模型通常指故障注入时机、位置、数目、宽度等。故障模型不同，产生的故障信息也不尽相同，对应的故障分析方法和结果也不同。需要说明的是故障攻击包括故障注入和故障分析 2 个部分，本文主要研究故障分析方法，故障注入方法不作为本文研究重点，读者可参考文献[17]。下面给出本文符号变换故障攻击模型。

如图 1 所示，滑动窗口算法的最大窗口宽度 w 是固定的，t 由当前待计算的密钥值 $k_j\sim k_{j-w+1}$ 决定，$t\leq w$。设一次点乘窗口滑动次数为 l，l 不大于密钥长度 n。为形式化表示公钥推算过程，用 P_{ui} 表示第 i 轮循环中 P_u 或 $-P_{-u}$ 的值，即当 $u>0$ 时，P_{ui} 表示 P_u，当 $u<0$ 时，P_{ui} 表示 $-P_{-u}$，则公钥可表示为

$$Q=2^{t_0}(\cdots 2^{t_{l-2}}(2^{t_{l-1}}Q_l+P_{u_{l-1}})+P_{u_{l-2}})\cdots)+P_{u_0} \quad (2)$$

从式(2)可知公钥计算过程中与私钥最相关的操作为算法 1 中的第③和④步，即图 1 中倍点运算（a）和加法运算（b），因此本文选取到（a）、(b) 2 处为故障注入位置；故障宽度为 1bit，仅改变符号位；故障数目由密钥分片大小和实验条件决定。下面分情况讨论不同故障位置和注入时机对密钥分析结果影响，以及具体的密钥恢复过程。

2.4　故障位于倍点运算的故障分析

2.4.1　故障分析

依据算法 1，首先根据当前待处理的密钥值 $k_j\sim k_{j-w+1}$ 计算得到 t_i（当 $k_j=0$ 时 $t=1$），然后执行倍点运算（a）操作。若在第 $i-1$ 轮（a）执行结束后注入一个符号变换故障，使得 $Q_i\neq Q_i'$，则产生的故障公钥可以表示为

$$\begin{aligned}R&=2^{t_0}(\cdots 2^{t_{i-1}}(Q_i'+P_{u_i})+P_{u_{i-1}})\cdots)+P_{u_0}\\&=-Q+2\sum_{j=0}^{i}2^{\sum_{l=0}^{j-1}t_l}P_{u_j}\end{aligned} \quad (3)$$

设 $U_i(P_u)=\sum_{j=0}^{i}2^{\sum_{l=0}^{j-1}t_l}P_{u_j}$，则

$$R=-Q+2U_i(P_u) \quad (4)$$

当窗口滑动次数为 i 且已计算的密钥长度为 v

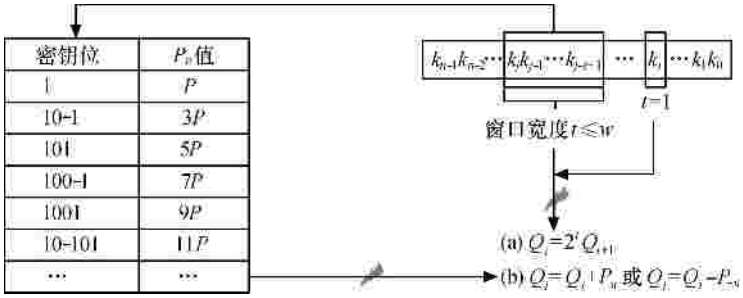

图 1　针对滑动窗口算法的故障模型

时，$U_i(P_u) = L_v(k)$，那么式(4)可以转化为

$$R=- Q+2L_v(k) \tag{5}$$

其中，$L_v(k) = \sum_{j=0}^{v} k_j 2^j P$。

可见由于故障的引入，导致故障公钥 R 的产生，并且通过变换得到的正确公钥 Q、故障公钥 R 之间的函数关系式（5）与文献[11]相同，可采用文献[11]中的密钥恢复算法进行密钥推算。算法 4 采用分而治之的思想逐片恢复密钥（假设每个片段中都注入故障），由于片段中故障位置的任意性，将分 2 种情况讨论其对密钥恢复过程的影响。

情况 1　故障注入时当前待处理的密钥位 k_j?0，则 u?0，需查找预计算表并对 Q_i 进行 2 次更新（乘法运算和加法运算）。

片段密钥恢复过程如图 2 所示：在 $n–i$ 轮注入故障，设 $Q + R= T_x$，由式(5)可知 $T_x = 2L_i(k)$。$k_s \sim k_0$ 表示已恢复密钥片段。恢复片段 X 的过程即寻找可能的 x_{s+r}~x_{s+1} 的 NAF_2 组合，使其满足 $T_x=X+2L_s$，其中，$X = 2\sum_{j=s+1}^{s+r} x_j 2^j P$，满足条件的 x_{s+r}~x_{s+1} 即为 k_j~k_{s+1}。

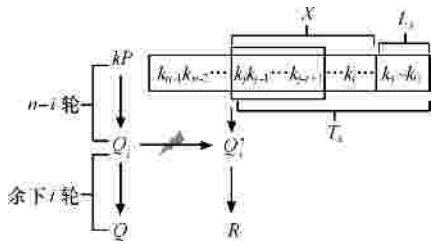

图 2　u?0 时片段密钥的恢复过程

情况 2　故障注入时当前待处理的密钥位 k_j=0，则 u=0，不需要查找预计算表，对 Q_i 只进行 1 次更新（只执行乘法运算）。

如图 3 所示，k_{z1}~k_{zr} 为 0，假设故障注入于 k_{zr} 处，使 Q_{i+1} 产生符号变化，则由式(2)可得故障公钥的表达式为

$$R_1 = 2^{\sum_{j=0}^{i-1} t_j} 2^{t_i} Q'_{i+1} + (2^{\sum_{j=0}^{i} t_j} P_{u_{i+1}} + \sum_{j=0}^{i} 2^{\sum_{l=0}^{j-1} t_l} P_{u_j})$$

由于 t^j=1，因此有

$$2^{t_i} Q'_{i+1} = 2Q'_{i+1} = -2Q_{i+1} = -Q_i = Q'_i \tag{6}$$

又因为 $P_{u_{i+1}} = 0$，$2^{\sum_{j=0}^{i} t_j} P_{u_{i+1}} = 0$，所以有

$$R_1 = 2^{\sum_{j=0}^{i-1} t_j} Q'_i + \sum_{j=0}^{i} 2^{\sum_{l=0}^{j-1} t_l} P_{u_j} = R \tag{7}$$

同样，如果将故障注入到 k_{z1}~k_{zr} 的任一位，均可得到 R_i=R，说明 u=0 时故障注入分析结果等效于下一个查表操作（u?0 时）注入故障。所以滑动窗口算法故障攻击中每次所恢复密钥长度并不是与故障位置严格对应，而是与故障影响的密钥位数密切相关。

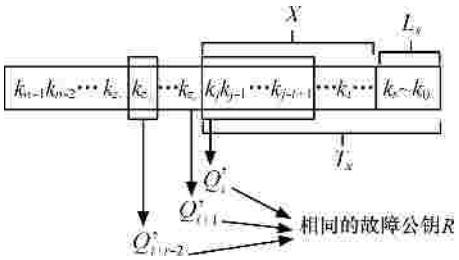

图 3　u=0 时片段密钥的恢复过程

2.4.2　针对“零块失效”的改进密钥恢复算法

文献[11]中密钥恢复算法中的“零块失效”(zero block failure）指密钥中存在长段的连续 0bit 所导致的攻击算法失效。这是由于相邻 2 个故障之间密钥位全是 0，使得 2 片故障信息完全相同而无法确定中间 0 的个数。本文将出现“零块”密钥与临近密钥片段合并，对文献[11]中的恢复算法加以改进。改进算法在出现“零块”时仍然可在有效时间内恢复完整密钥。

算法 2　存在“零块失效”的片段密钥恢复算法

输入：“零块失效”出现时已恢复的密钥片段 k_0~k_s，“零块失效”片段故障 R_1，“零块失效”片段下一段故障 R_2，正确公钥 Q；

输出：$k_{s+1} \sim k_{s+w+r}$；

1) 计算 $L = 2\sum_{j=0}^{s} k_j 2^j P$；

2) $w = 1$；

3) while ($w < 2^m$) ；

4) for(所有的长度 r = 1, 2,…,m)do；

5)　for(所有的二进制组合 x =(x_{s+1}, x_{s+2},…, x_{s+r}))do；

6)　$T_x = L + 2\sum_{j+w+1}^{s+w+r} x_j 2^j P$；

7)　if ($Q + R_2 = T_x$) then；

8)　$k_{s+1} \sim k_{s+w} = 0$；

k_{s+w+1}=x_{s+w+1}, k_{s+w+2}=x_{s+w+2},…,k_{s+w+r}=x_{s+w+r}；

9)　else $w = w + 1$。

2.5　故障位于加法运算的故障分析

图 1 中加法操作(b)只有在当前待计算的密钥位 k_j?0 时才执行，并非任意位置都可以注入故障，所以本节讨论情形在一定程度依赖于密钥数值。下面就故障注入于运算(b)中 2 个不同变量分别讨论。

2.5.1　针对变量 P_u 的故障分析

图 1 中 k_j?0 时，需先执行查表操作再执行加法运算。文中假设故障注入到查表操作之后，令 P_u 产生符号变换故障，即 P_{ui}? P_{ui}'，产生故障信息为

$$R = 2^{t_0}(\text{L}\,2^{t_{i-1}}(2^{t_i}Q_{i+1} + P_{u_i}^{'}) + P_{u_{i-1}})\,\text{L}) + P_{u_0}$$
$$= Q - 2^{1+\sum_{j=0}^{i-1} t_j} P_{u_i} \tag{8}$$

从式(8)可知 R 仅与查表索引和故障密钥位置 j 相关，因此一个故障仅能恢复一个窗口密钥，下面给出故障位于 P_u 时的密钥恢复算法。

算法 3　故障位于 P_u 的密钥恢复算法

输入：最大窗口数 w，正确公钥 Q，故障公钥 R；

输出：密钥 k；

1) 设集合 S={R|R 表示故障公钥}；

2) 初始化密钥 k_{n-1}~k_0=0；

3) 根据 w 计算 X={xP|x=P,3P,…,2(2^w- (- 1)^w/3- 1)P}；

4) for（S 中每一个元素）；

5)　for（i=0; i<n; i++）；

6)　for（X 中的每一个元素 xP）；

7)　if（2^ixP+R=Q 或- 2^ixP+R=Q）；

8)　则 k_j~k_r =$NAF_2(x)$ ；

9) if（Q=kP），输出 k。

对于一次特定的故障攻击，最理想的情况是每次查找预计算表都能成功注入一个符号变换故障，利用算法 3 可成功恢复完整密钥。但实际情况下，故障往往不能精确注入到每次滑动，此时算法 3 仅能恢复产生故障的密钥片段，需结合格攻击恢复完整密钥。格攻击所需条件是已知三元组（s_i,m_i,k_i），即签名、消息、部分私钥(可根据公钥获取到签名)[18]。Bob[15]等人通过监视 Cache 中预计算表的访存时间差异，利用 Cache 旁路模板信息结合隐马尔科夫模型，获取到部分滑动窗口中的密钥值，在不到一小时的时间恢复了 160bit 完整密钥。这与本节的情形是一致的，具体细节请参阅文献[15]。

2.5.2　针对变量 Q_i 的故障分析

加法操作（b）中的 Q_i 也是可能的故障注入点，其故障分析类似于下一轮循环时（a）操作注入故障的分析过程，这里不再赘述。二者区别是：2.4 节中将故障注入于（a）处，恢复的密钥位为图 4 中 k_i~k_{s+1}，即从故障注入位至已恢复密钥位；2.5 节故障位于（b）处时，所恢复密钥位为图中 k_j~k_{s+1}，即从故障位的下一窗口（非零窗口）至已恢复密钥位。

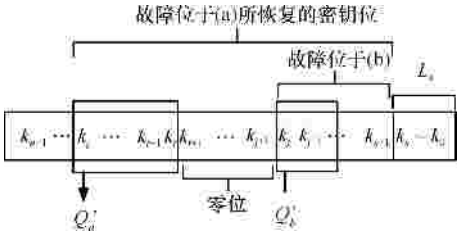

图 4　故障位于（a）、(b）处恢复密钥位区别

3　攻击实验

在普通 PC 机（CPU 为 Athlon 64 3000+ 1.81GHz，内存为 1GB）上使用 C++语言（Visual C++ 6.0 环境）编程实现了本文 ECC 故障攻击，其中，故障诱导是通过软件模拟的。

利用 2.4 节故障分析方法，对 OpenSSL0.9.8a 中 SECG 和 NIST X9.62 提供的 8 条安全椭圆曲线进行了攻击仿真实验，密钥长度为 192bit，故障样本数分别为 30、20、15 这 3 种。

实验结果如图 5 所示，结果表明以下 3 个结论。

1) 在相同曲线条件下，3 组实验的攻击时间有明显不同。攻击时间差异是由于密钥片段大小 m 决定了搜索空间 3^m，故障样本数越大，密钥片段越小，攻击时间越短。

2) 在相同故障样本下，11 条曲线的攻击时间有明显不同。攻击时间差异主要是由点乘运算开销 M 所引起的，即基域越大，点乘运算开销越大。

3) 基域相同的 3 条曲线（NIST 192、NIST 192v2、NIST 192v3）攻击时间差异不大。这是由于相同基域下的曲线参数 a、b 和基点对点乘运算的影响不大。

由于 OpenSSL0.9.8a 中自动生成的密钥不存在大段的连续的零，为验证算法 2，将密钥中的一段置零，再对 NIST 192 3 条曲线进行故障攻击（样本量选取 20）。当出现“零块”时调用算法 2，攻击时间如图 5 所示：攻击时间比不存在“零块”的密钥多耗时 70s 左右，多余耗时主要消耗在合并密钥

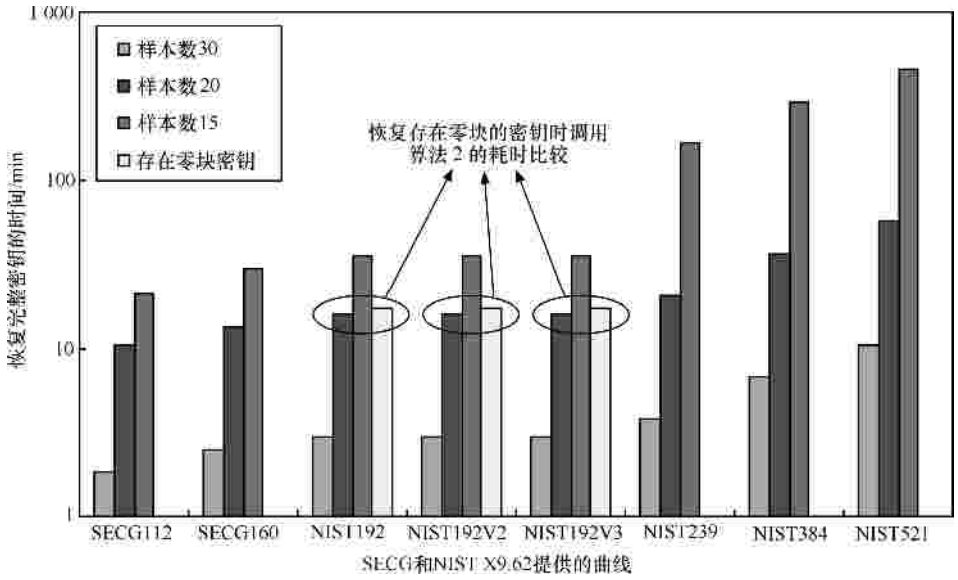

图 5 针对不同曲线参数恢复 192bit 完整密钥所需时间

表 1 针对不同密钥、不同故障样本的密钥搜索空间

私钥 k	N	L	F	O
3A3806B26502C8809	191	30	15	$3^{107.06}$
6EDC5EA156C7934A			20	$3^{76.32}$
3EBFA6F39C1044E			30	$3^{10.69}$
7A2C5B5B27D2D17F	192	32	15	$3^{118.06}$
81EF7315CE9EA059C			20	$3^{87.32}$
19E52CAD68B909D			30	$3^{22.69}$
ABA19002453CC5BCD	193	31	15	$3^{94.07}$
ACDCA91FC8D74C9D			20	$3^{72.33}$
5B1C5D6F6AA3AF9			30	$3^{13.70}$

注：N 为密钥 NAF_2 型表示的长度；L 为窗口滑动次数；F 为故障个数；O 为密钥搜索空间。

块的穷举上。同时作者发现出现“零块”的位置与攻击时间也密切相关，即出现“零块”时已知的密钥位越多，算法 2 进行穷举时点乘开销越大，完成攻击所需时间越多。

利用 2.5 节中针对 P_u 的故障分析方法，针对 NIST 192 的 3 个不同私钥 k 进行故障攻击，表 1 给出了不同私钥、不同故障数样本数的密钥搜索空间。在每次滑动都能引入故障的情况下，恢复完整密钥仅需 1min；当故障不能精确注入到每一次滑动时，1 个故障大约可以将密钥空间降低 2^7~2^{15}。Römer[19]证明了在已知 12bit 的密钥和 50 个签名的情况下，格攻击成功率可达到 99%。所以利用算法 3 结合攻击，3~5 个故障样本即可恢复完整密钥。

现有 ECC 故障攻击研究大都在仿真环境下进行，且已发表文献中尚未有针对滑动窗口算法的 ECC 故障攻击。基于符号变换的 ECC 故障攻击只有一例[12]，主要针对采用 NAF_2 实现点乘的 ECC，文中也并没有给出具体攻击时间。本文针对滑动窗口算法实现的 ECC 进行了故障攻击仿真实验，给出了一种可消除了“零块失效”问题的改进算法，通过改变故障注入点提出了一种新的故障分析方法，对不同基域下的 ECC 进行了攻击仿真实验，并给出了恢复完整密钥所需时间。由于点乘运算实现方式中都存在倍点运算，故 2.4 节的故障分析方法也适用于其他采用点乘运算的

密码算法。

4　结束语

在故障攻击方面主要有 3 个研究方向：故障注入、故障分析、故障攻击检测与防护，本文主要对滑动窗口算法的椭圆曲线密码故障分析进行了研究，并通过软件仿真进行了验证。以下方面值得在将来进行后续研究和关注：第一，研究 FPGA 上的故障注入方法，利用文中故障分析方法，开展对 ECC 故障攻击物理实验；第二，研究 ECC 故障攻击检测与防护措施。

参考文献：

[1] KOEUNE F, STANDAERT F X. A tutorial on physical security and side-channel attacks[A]. Foundations of Security Analysis and Design III: FOSAD 2004/2005 Tutorial Lectures[C]. Forli, Italy, 2005. 78-108.

[2] BONEH D, DEMILLO R, LIPTON R. On the importance of checking cryptographic protocols for faults[A]. Eurocrypt 1997[Konstanz, Germany, 1997. 37-51.

[3] MUKHPOADHYAY D. An improved fault based attack of the advanced encryption standard[A]. AFRICACRYPT 2009[C]. Gammarth, Tunisia, 2009. 421-434.

[4] **李玮**, **谷大武**. **基于密钥编排故障的**SMS4 **算法的差分故障分析**[J]. **通信学报**, 2008, 29(10): 135-142.
LI W, GU D W. Differential fault analysis on the SMS4 cipher by schedule[J]. Journal on Communications, 2008, 29(10): 135-142.

[5] **赵新杰**, **王韬**, **王素贞等**. **针对** MIBS **的深度差分故障分析**[J]. **通信学报**, 2010, 31(12): 82-89.
ZHAO X J, WANG T, WANG S Z, *et al*. Research on deep differntial fault analysis against MIBS[J]. Journal on Communications, 2010, 31(12): 82-89.

[6] BIHAM E, GRANBOULAN L, NGUYN P Q. Impossible fault analysis of RC4 and differential fault analysis of RC4[A]. E 2005[C]. Lisbon, Portugal, 2005. 359-367.

[7] CORON J S. Resistance against differential power analysis for elliptic curve cryptosystems[A]. CHES 1999[C]. Massachusetts, USA, 1999. 292-302.

[8] BIEHL I, MEYER B, MLLER V. Differential fault attacks on elliptic curve cryptosystems[A]. CRYPTO 2000[C]. Berlin, Germany, 2000. 131-146.

[9] ANTIPA A, DANIEL B, MENEZES A, *et al*. Validation of elliptic curve public keys[A]. PKC 2003[C]. Miami, USA, 2003. 211-223.

[10] FOUQUE P A, LERCIER R. Fault attack on elliptic curve with montgomery ladder implementation[A]. FDTC 2008[C]. Washington DC, USA, 2008. 92-98.

[11] BLOMER J, OTTO M, SEIFERT J P. Sign change fault attacks on elliptic curve cryptosystems[A]. FDTC 2006[C]. Yokohama, Japan, 2006. 36-52.

[12] SCHMIDT J M, MEDWED M. A fault attack on ECDSA[A]. FDTC 2009[C]. Lausanne, Switzerland, 2009. 93-99

[13] EBEID N, LAMBERT B. Securing the elliptic curve montgomery ladder against fault attacks[A]. FDTC 2009[C]. Lausanne, Switzerland, 2009. 46-50.

[14] **王潮**, **时向勇**,**牛志华**. **基于** Montgomery **曲线改进** ECDSA **算法的研究**[J]. **通信学报**, 2010, 31(1): 9-13.
WANG C, SHI X Y, NIU Z H. The research of the promotion for ECDSA algorithm based on Montgomery-form ECC[J]. Journal on Communications, 2010, 31(1): 9-13.

[15] BRUMLEY B B, RISTO M H. Cache-timing template attacks[A]. ASIACRYPT 2009[C]. Tokyo, Japan, 2009. 667-684.

[16] REDDY E K. Elliptic curve cryptosystems and side-channel attacks[J]. International Journal of Network Security, 2011, 12(3): 151-158.

[17] GIRAUD C, THIEBEAULD H. A survey on fault attacks[A]. CARDIS 2004[C]. Toulouse, France, 2004. 22-27.

[18] SMART N P, GRAHAM N H. Lattice attacks on digital signature

schemes[J]. Codes and Cryptography, 2001, 23(3): 283-290.

[19] ROMER T, SEIFERT J P. Information leakage attacks against smart card implementations of the elliptic curve digital signature algorithm[A]. E-smart 2001[C]. Cannes, France, 2001. 211-219.

作者简介：

张金中（1985-），男，辽宁沈阳人，军械工程学院硕士生，主要研究方向为网络安全技术。

寇应展（1964-），男，陕西临潼人，军械工程学院教授、硕士生导师，主要研究方向为网络安全技术。

王韬（1964-），男，河北石家庄人，军械工程学院教授、博士生导师，主要研究方向为信息安全和密码旁路分析。

郭世泽（1969-），男，河北石家庄人，北方电子设备研究所研究员，主要研究方向为信息安全和密码学。

赵新杰（1986-），男，河南开封人，军械工程学院博士生，主要研究方向为分组密码旁路分析和故障分析。