西安電子科技大學
XIDIAN UNIVERSITY

LadderLeak and Bleichenbacher’s HNP Attack

LadderLeak: Breaking ECDSA With Less Than One Bit Of Nonce Leakage (CCS 2020)

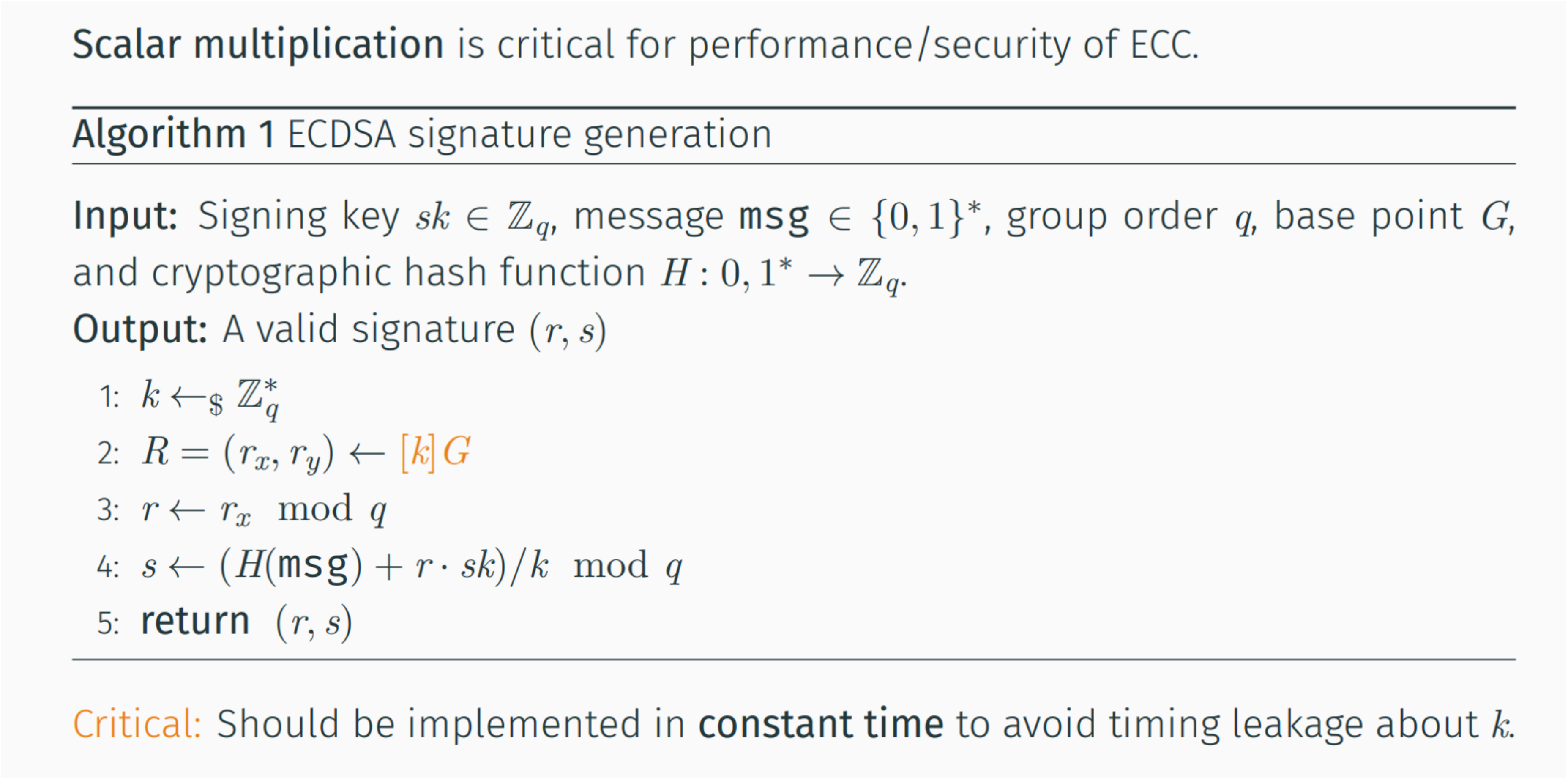
主讲：随缘 (su1yu4n@gmail.com)

总览

- 预备知识：ECDSA 和 HNP问题
- 攻击效果与论文贡献介绍
- LadderLeak漏洞
- **Bleichenbacher攻击框架**
- Bleichenbacher攻击细节及其优化

预备知识

ECDSA signature



Scalar multiplication is critical for performance/security of ECC.

Algorithm 1 ECDSA signature generation

Input: Signing key $sk \in \mathbb{Z}_q$, message $\mathtt{msg} \in \{0,1\}^*$, group order q, base point G, and cryptographic hash function $H : 0, 1^* \rightarrow \mathbb{Z}_q$.

Output: A valid signature (r, s)

```
$k \leftarrow_{\$} \mathbb{Z}_q^*$
$R = (r_x, r_y) \leftarrow [k]G$
$r \leftarrow r_x \mod q$
$s \leftarrow (H(\mathtt{msg}) + r \cdot sk)/k \mod q$
return $(r, s)$
```

Critical: Should be implemented in **constant time** to avoid timing leakage about k.

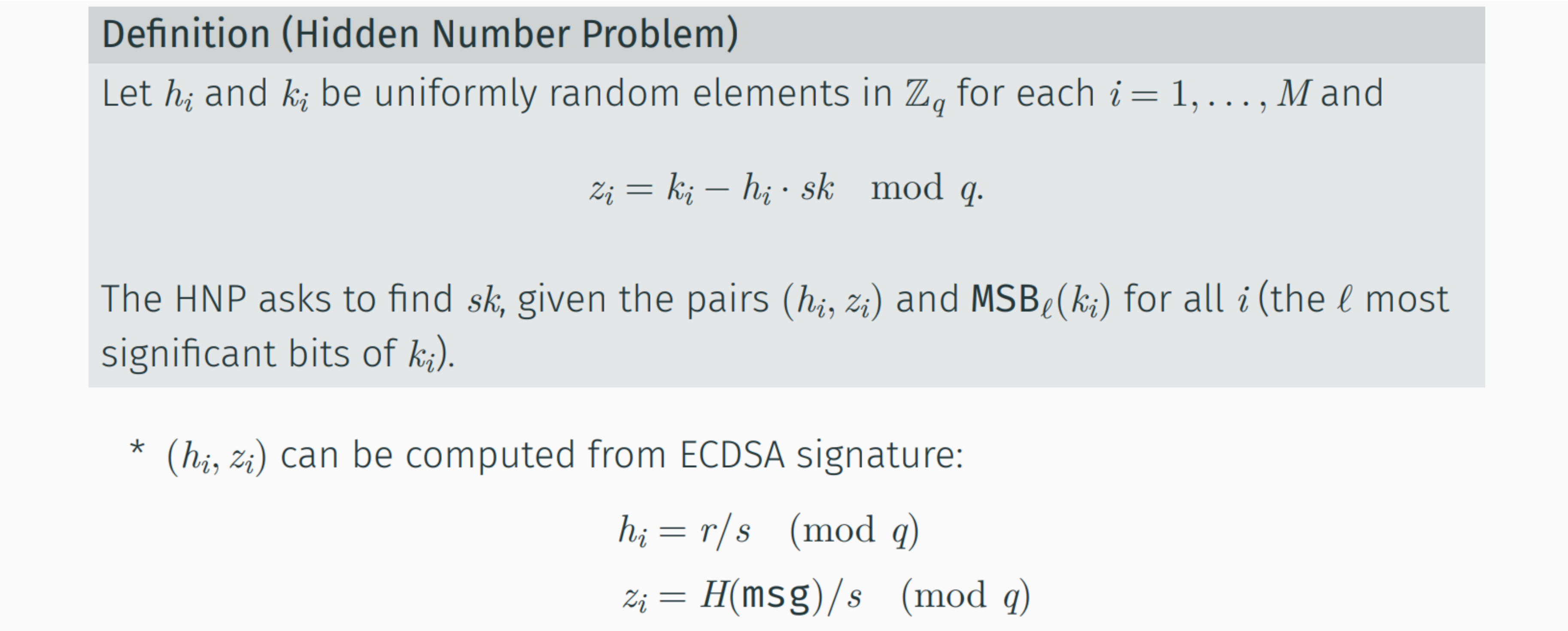
这里的 k 我们称为 nonce

nonce的安全性

Nonce的生成必须使用安全的RNG，要求：

- 不可泄漏
- 不可重用
- 不可太小(eg. 32bit / 64bit)
- **必须采样自均匀分布**

HNP – The Hidden Number Problem



Definition (Hidden Number Problem)

Let h_i and k_i be uniformly random elements in $\mathbb{Z}_q$ for each $i = 1, \ldots, M$ and

$$z_i = k_i - h_i \cdot sk \mod q.$$

The HNP asks to find sk, given the pairs (h_i, z_i) and $\mathsf{MSB}_\ell(k_i)$ for all i (the ℓ most significant bits of k_i).

* (h_i, z_i) can be computed from ECDSA signature:

$$h_i = r/s \pmod q$$
$$z_i = H(\mathtt{msg})/s \pmod q$$

ECDSA：$s_i = (H(m_i) + r_i \cdot sk)/k_i \mod q$

$\Rightarrow H(m)/s_i = k_i - (r_i/s_i) \cdot sk \mod q$

求私钥就是求解HNP问题。

两种针对HNP的攻击

基于格的攻击：

- 需要的签名数量少
- 每个nonce要泄露至少4比特
- 不能有错误数据（然而侧信道中泄露的比特往往有少量错误）

基于傅里叶分析的Bleichenbacher攻击：

- 需要的签名相对多
- 每个nonce只要泄露1~3比特（越多越好）
- 允许一定的错误数据（eg. 1%）

攻击效果与论文贡献

攻击效果

以 P-192 曲线为例，每个nonce k_i有几个MSB的泄露：

- 3bit leak: 获取几百个或几千个签名即可解出私钥
- 2bit leak: 几千个或几万个
- 1bit leak: 数百万个

如果是sect163r1，那么如果有数百万个签名，即使“小于一比特泄露”也能求出私钥。

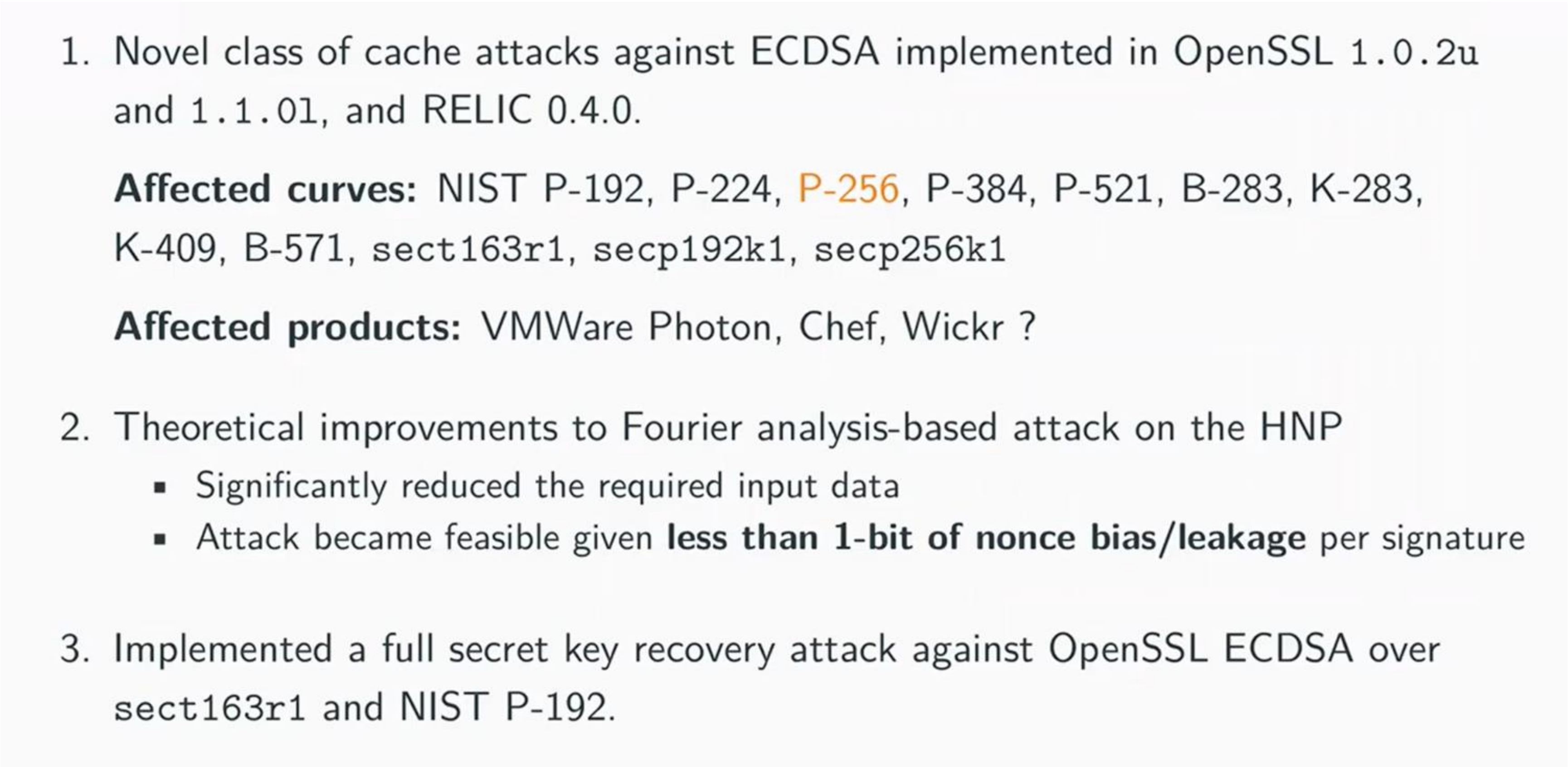
论文贡献

1. 发现了OpenSSL标量乘算法中的侧信道漏洞——LadderLeak，并利用LadderLeak泄露“小于一比特”

2. 改进了Bleichenbacher’s HNP attack的理论分析，改进了其中HGJ算法的使用方法。

3. 根据理论分析优化了HNP求解的攻击方法和实现，打破了对ECDSA攻击的记录。

论文贡献



1. Novel class of cache attacks against ECDSA implemented in OpenSSL 1.0.2u and 1.1.0l, and RELIC 0.4.0.

   Affected curves: NIST P-192, P-224, P-256, P-384, P-521, B-283, K-283, K-409, B-571, sect163r1, secp192k1, secp256k1

   Affected products: VMWare Photon, Chef, Wickr ?

2. Theoretical improvements to Fourier analysis-based attack on the HNP
   - Significantly reduced the required input data
   - Attack became feasible given **less than 1-bit of nonce bias/leakage** per signature

3. Implemented a full secret key recovery attack against OpenSSL ECDSA over sect163r1 and NIST P-192.

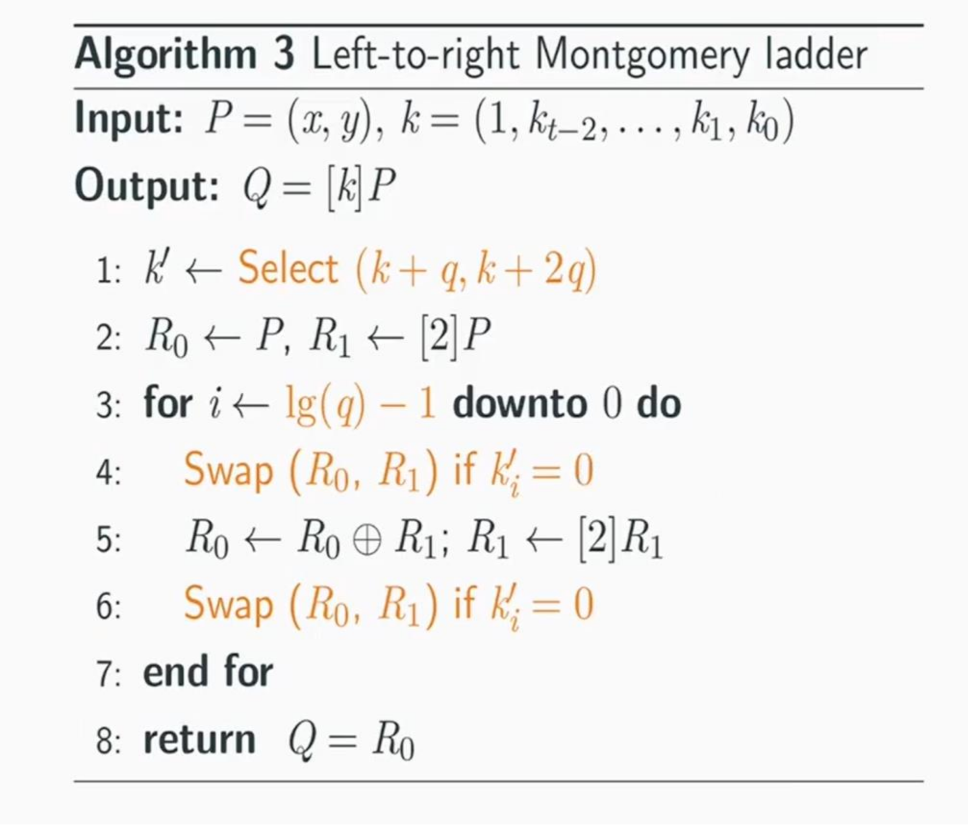
LadderLeak漏洞简介

标量乘算法

时间侧信道攻击：通过计算时间推测出敏感信息（密钥、明文等相关信息）

- 决不能使用课本中的Double and Add算法（类似模平方）
- 实际通常采用侧信道安全的Montgomery Ladder算法

安全的Montgomery Ladder算法



```
Algorithm 3 Left-to-right Montgomery ladder
Input: $P = (x, y)$, $k = (1, k_{t-2}, \dots, k_1, k_0)$
Output: $Q = [k]P$
  $k' \leftarrow$ Select $(k + q, k + 2q)$
  $R_0 \leftarrow P$, $R_1 \leftarrow [2]P$
  for $i \leftarrow \lg(q) - 1$ downto $0$ do
      Swap $(R_0, R_1)$ if $k'_i = 0$
      $R_0 \leftarrow R_0 \oplus R_1$; $R_1 \leftarrow [2]R_1$
      Swap $(R_0, R_1)$ if $k'_i = 0$
  end for
  return $Q = R_0$
```

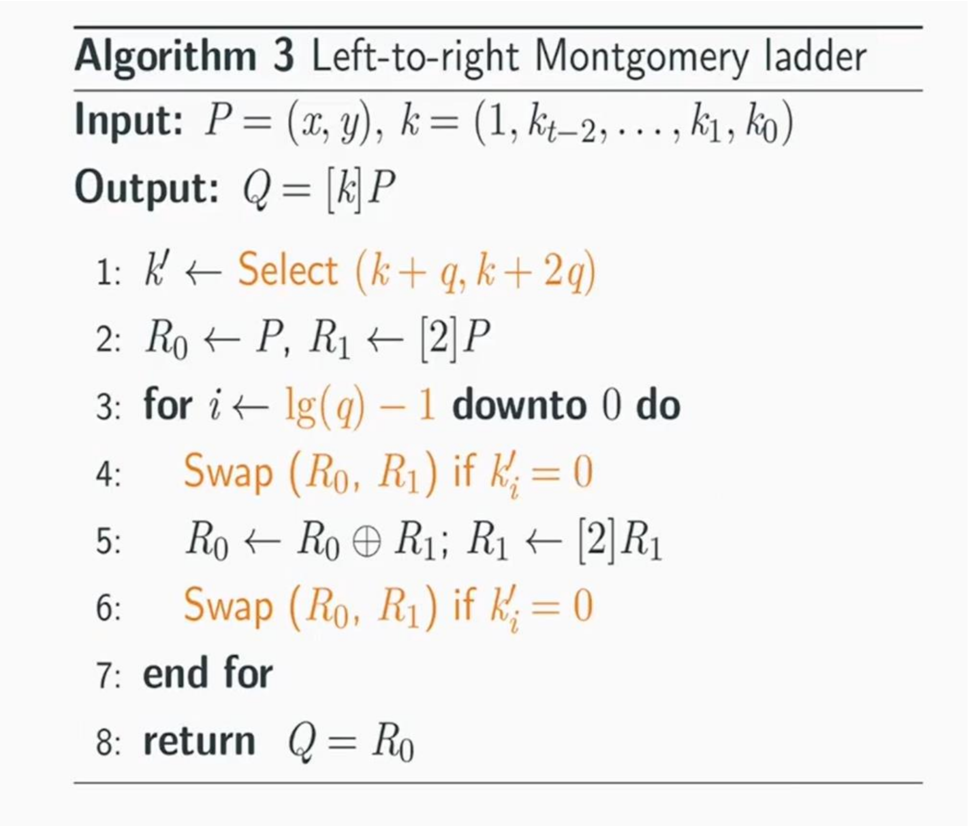
安全的算法需要满足三条：

1. 循环迭代次数必须固定；

2. 内存中的运算不能依赖于秘密标量nonce的位，以避免通过内存层次结构泄漏（如缓存泄露）；

3. 不论标量的值是什么，做加法时必须以相同的顺序进行固定的数量和**固定类型的运算（OpenSSL的实现不满足这一点）**

LadderLeak漏洞



```
Algorithm 3 Left-to-right Montgomery ladder
Input: $P = (x, y)$, $k = (1, k_{t-2}, \dots, k_1, k_0)$
Output: $Q = [k]P$
1: $k' \leftarrow$ Select $(k + q, k + 2q)$
2: $R_0 \leftarrow P$, $R_1 \leftarrow [2]P$
3: for $i \leftarrow \lg(q) - 1$ downto $0$ do
4:     Swap $(R_0, R_1)$ if $k'_i = 0$
5:     $R_0 \leftarrow R_0 \oplus R_1$; $R_1 \leftarrow [2]R_1$
6:     Swap $(R_0, R_1)$ if $k'_i = 0$
7: end for
8: return $Q = R_0$
```

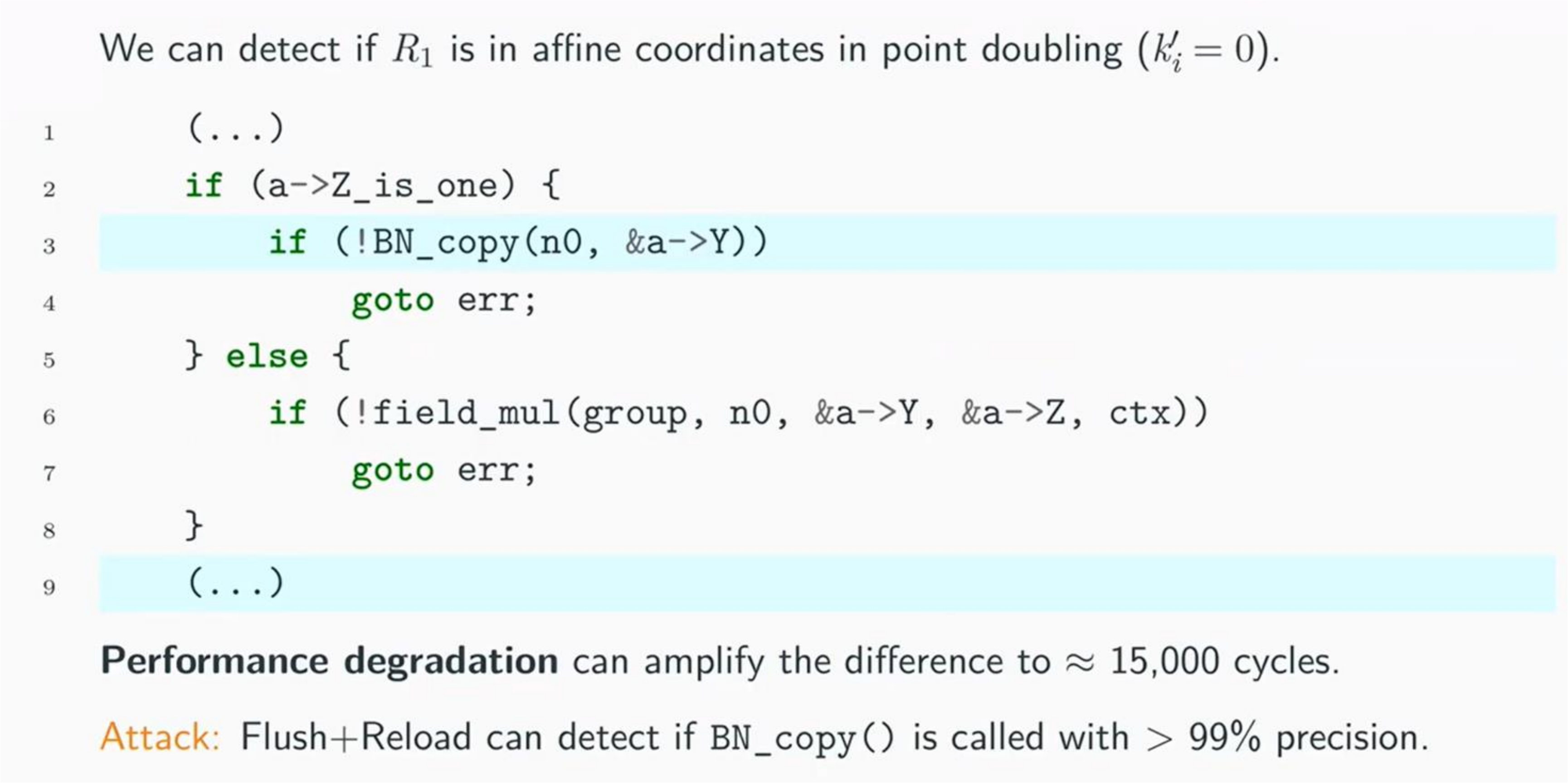
以OpenSSL 1.0.2版本为例。**算法的第二步R_0采用仿射坐标，而R_1却采用了射影坐标。两种坐标下计算速度略有不同。**第5步中计算$R_1 \leftarrow [2]R_1$时能用**缓存计时攻击看第五步执行的时间，**推断出此时的R_1的坐标类型，进而得知第4步是否执行了Swap语句，也就得知了与nonce最高位相关的值k_i'。最终的攻击效果是能够**以极大的概率正确获取nonce的一个MSB**。

缓存计时攻击

时间侧信道攻击：通过计算时间推测出敏感信息（密钥、明文等相关信息）

- 假定：攻击者在受害服务器上运行自己的一个“木马”，并且也使用同样版本的密码库。
- 攻击方法：监视内存中某一段指令A，首先执行CLFLUSH汇编指令清掉缓存中的指令A 。然后很快重新访问指令A，如果取指令A的速度很快意味着受害者执行过A，如果慢说明受害者没执行过A

缓存计时攻击



We can detect if R_1 is in affine coordinates in point doubling $(k'_i = 0)$.

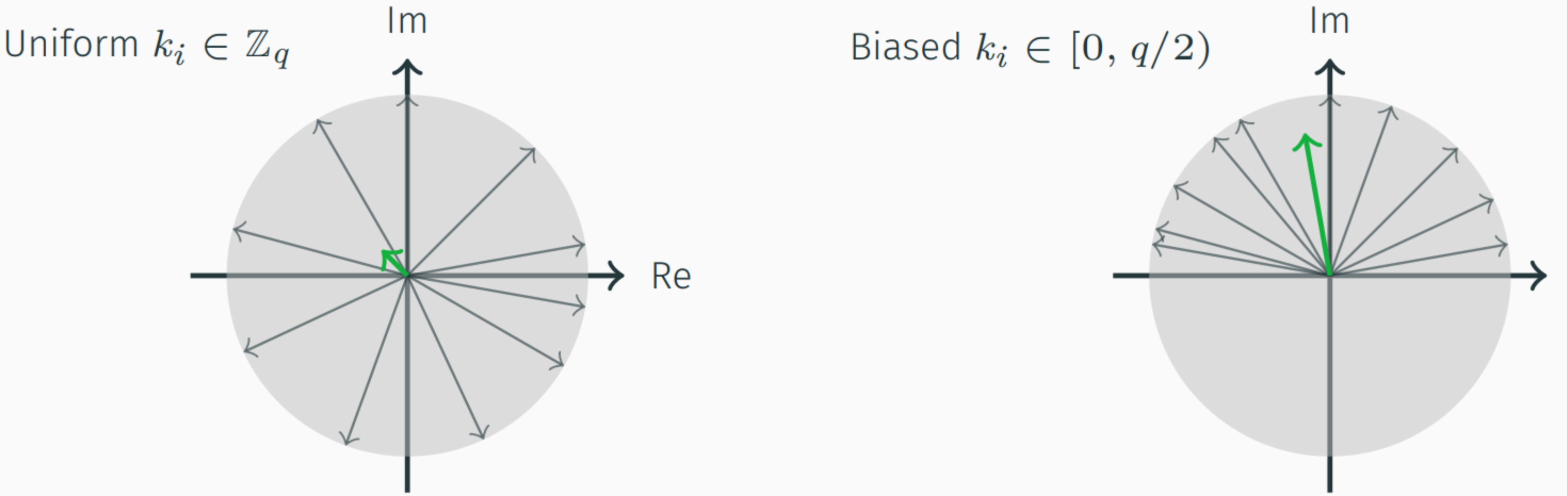
```
(...)
if (a->Z_is_one) {
    if (!BN_copy(n0, &a->Y))
        goto err;
} else {
    if (!field_mul(group, n0, &a->Y, &a->Z, ctx))
        goto err;
}
(...)
```

Performance degradation can amplify the difference to $\approx$ 15,000 cycles.

Attack: Flush+Reload can detect if `BN_copy()` is called with $> 99\%$ precision.

Bleichenbacher攻击框架

直观看Bias函数



Definition

The **sampled bias** of a set of points $K = \{k_i\}_{i \in \{1,\dots,M\}}$ in $\mathbb{Z}_q$ is defined by

$$\text{Bias}_q(K) = \frac{1}{M} \sum_{i=1}^{M} e^{2\pi \mathrm{i} k_i / q}.$$

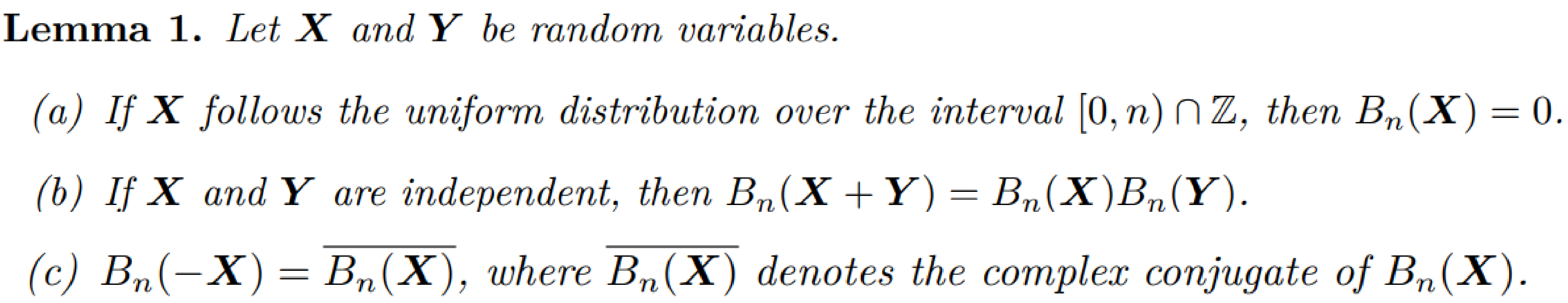
Bias函数

- 令$\boldsymbol{X}$为$\mathbb{Z}_q$上的随机变量，则bias函数$B_q(\boldsymbol{X})$ 定义为

$$B_q(\boldsymbol{X}) = \mathbf{E}\left(\mathrm{e}^{2\pi\mathrm{i}\boldsymbol{X}/q}\right) = B_q(\boldsymbol{X} \bmod q)$$

- 类似地，对于 $V = \{v_j\} \subset \mathbb{Z}_q$, $B_q(V)$ **定义为一个DFT：**

$$B_q(V) = \frac{1}{n}\sum_{j=0}^{n-1} e^{2\pi\mathrm{i}v_j/q}$$



Lemma 1. *Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ be random variables.*

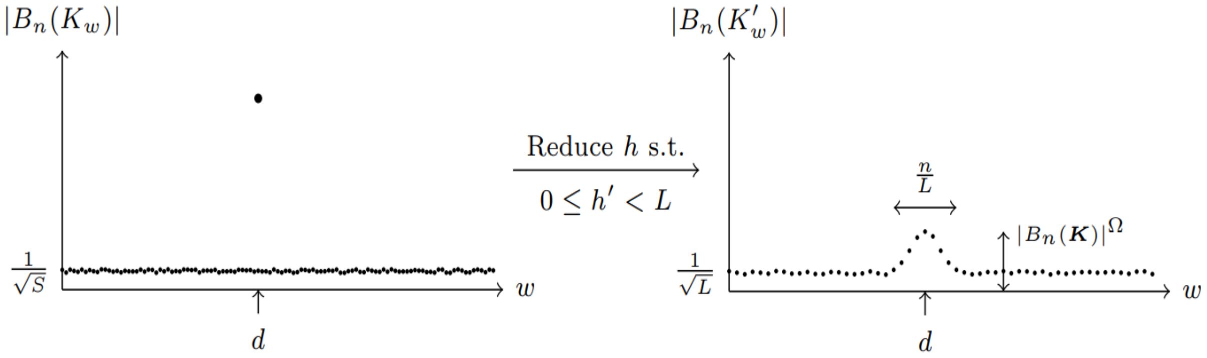
(a) If $\boldsymbol{X}$ follows the uniform distribution over the interval $[0, n) \cap \mathbb{Z}$, then $B_n(\boldsymbol{X}) = 0$.

(b) If $\boldsymbol{X}$ and $\boldsymbol{Y}$ are independent, then $B_n(\boldsymbol{X} + \boldsymbol{Y}) = B_n(\boldsymbol{X})B_n(\boldsymbol{Y})$.

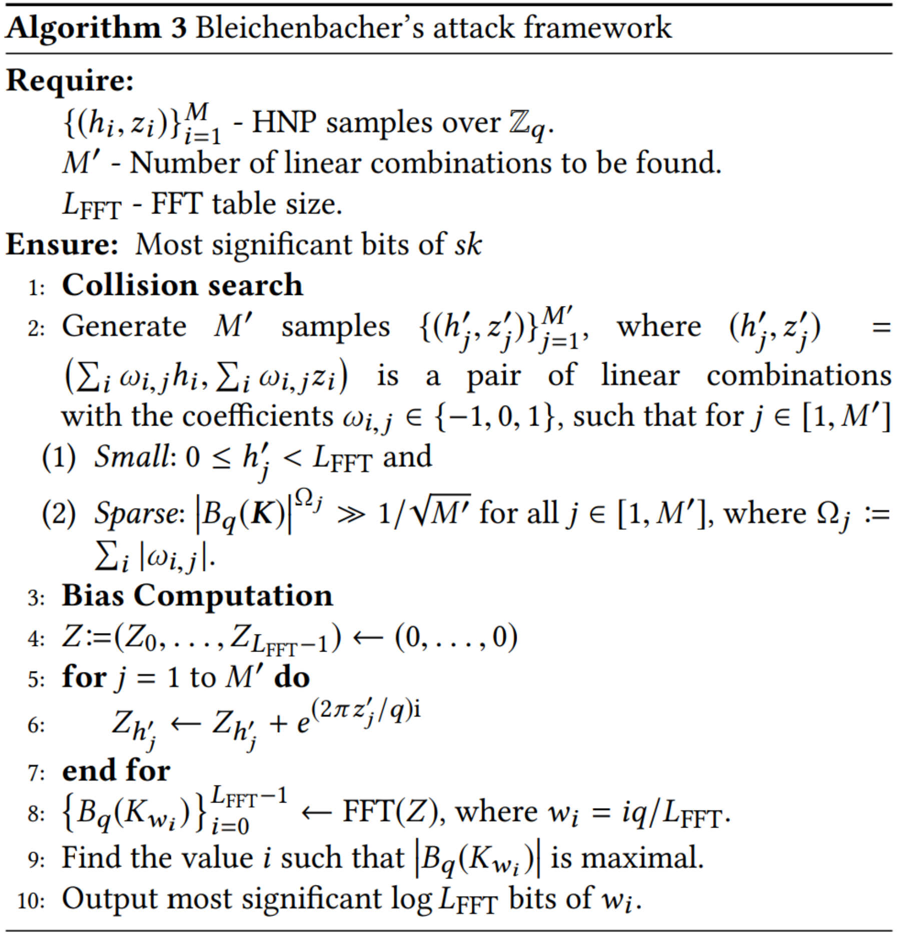
(c) $B_n(-\boldsymbol{X}) = \overline{B_n(\boldsymbol{X})}$, where $\overline{B_n(\boldsymbol{X})}$ denotes the complex conjugate of $B_n(\boldsymbol{X})$.

用bias函数区分密钥

- 对HNP等式变形可得 $k_i = z_i + h_i \cdot sk \bmod q$.
- 攻击原理：**bias函数在正确的nonce处取得最大值，据此可找出密钥**。设$w \in \mathbb{Z}_q, K_w = \{z_i + h_i w\}_{i=0}^{S-1}$，$w = sk$时$|B_q(K_w)|$接近1，而$w \neq sk$时$B_q(K_w)$接近$1/\sqrt{S}$. **因此我们只需找到函数$|B_q(K_w)|$的最值点**。
- 穷举$sk \in \mathbb{Z}_q$不可取，但range reduction让函数图像从冲激变缓坡。



Bleichenbacher攻击框架



```
Algorithm 3 Bleichenbacher's attack framework
Require:
    $\{(h_i, z_i)\}_{i=1}^{M}$ - HNP samples over $\mathbb{Z}_q$.
    $M'$ - Number of linear combinations to be found.
    $L_{\mathrm{FFT}}$ - FFT table size.
Ensure: Most significant bits of $sk$
    Collision search
    Generate $M'$ samples $\{(h'_j, z'_j)\}_{j=1}^{M'}$, where $(h'_j, z'_j) = (\sum_i \omega_{i,j} h_i, \sum_i \omega_{i,j} z_i)$ is a pair of linear combinations with the coefficients $\omega_{i,j} \in \{-1, 0, 1\}$, such that for $j \in [1, M']$
      (1) Small: $0 \le h'_j < L_{\mathrm{FFT}}$ and
      (2) Sparse: $|B_q(\mathbf{K})|^{\Omega_j} \gg 1/\sqrt{M'}$ for all $j \in [1, M']$, where $\Omega_j := \sum_i |\omega_{i,j}|$.
    Bias Computation
    $Z := (Z_0, \dots, Z_{L_{\mathrm{FFT}}-1}) \leftarrow (0, \dots, 0)$
    for $j = 1$ to $M'$ do
        $Z_{h'_j} \leftarrow Z_{h'_j} + e^{(2\pi z'_j/q)i}$
    end for
    $\{B_q(K_{w_i})\}_{i=0}^{L_{\mathrm{FFT}}-1} \leftarrow \mathrm{FFT}(Z)$, where $w_i = iq/L_{\mathrm{FFT}}$.
    Find the value $i$ such that $|B_q(K_{w_i})|$ is maximal.
    Output most significant $\log L_{\mathrm{FFT}}$ bits of $w_i$.
```

1. **Range reduction（关键）**
2. 用FFT计算Bias函数
3. 比较大小算出sk

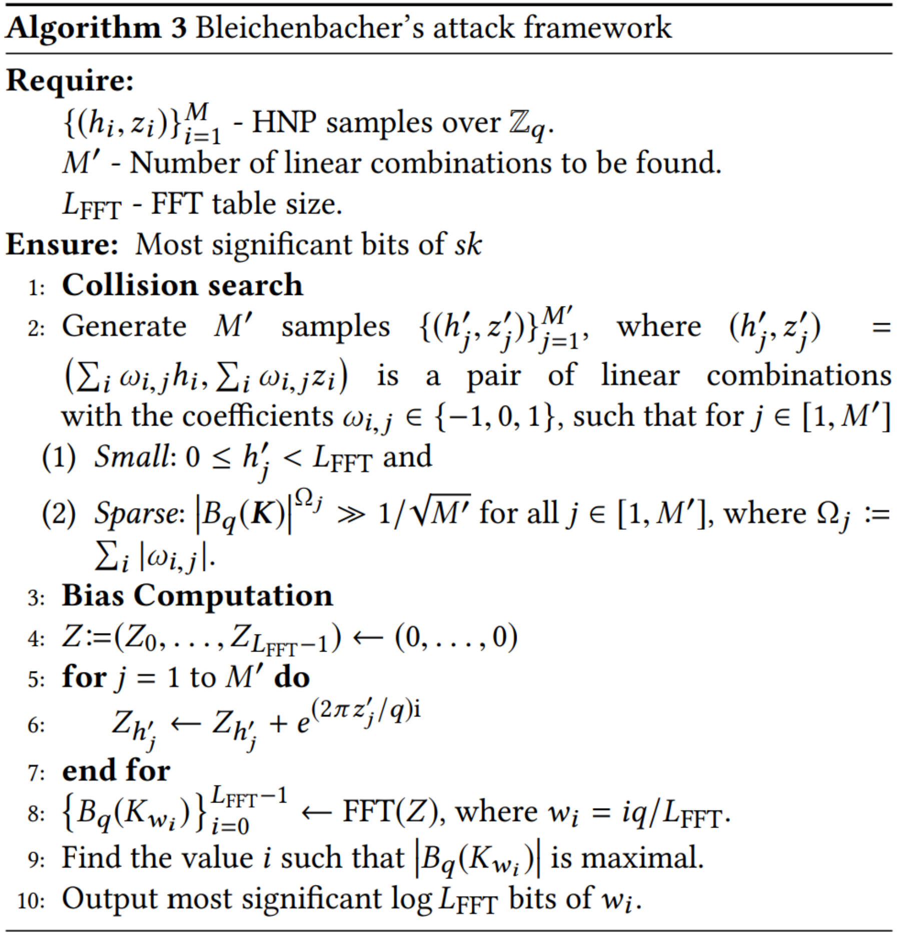
Range reduction: 取(h_i, z_i)的小系数线性组合(h'_i, z'_i)，(h'_i, z'_i)满足small和sparse

Small: 让h'_i变小，图像变成缓坡

Sparse: 取线性组合的**副作用是峰值变小**。为了分辨出峰值，线性组合系数**大部分系数为0**.

（设随机变量$\boldsymbol{K}_i$对应nonce k_i，相加减后Bias函数模值减小）

Bleichenbacher攻击框架



Algorithm 3 Bleichenbacher's attack framework

```
Require:
    $\{(h_i, z_i)\}_{i=1}^{M}$ - HNP samples over $\mathbb{Z}_q$.
    $M'$ - Number of linear combinations to be found.
    $L_{\mathrm{FFT}}$ - FFT table size.
Ensure: Most significant bits of $sk$
Collision search
Generate $M'$ samples $\{(h'_j, z'_j)\}_{j=1}^{M'}$, where $(h'_j, z'_j) = (\sum_i \omega_{i,j} h_i, \sum_i \omega_{i,j} z_i)$ is a pair of linear combinations with the coefficients $\omega_{i,j} \in \{-1, 0, 1\}$, such that for $j \in [1, M']$
  (1) Small: $0 \le h'_j < L_{\mathrm{FFT}}$ and
  (2) Sparse: $|B_q(\mathbf{K})|^{\Omega_j} \gg 1/\sqrt{M'}$ for all $j \in [1, M']$, where $\Omega_j := \sum_i |\omega_{i,j}|$.
Bias Computation
$Z := (Z_0, \ldots, Z_{L_{\mathrm{FFT}}-1}) \leftarrow (0, \ldots, 0)$
for $j = 1$ to $M'$ do
    $Z_{h'_j} \leftarrow Z_{h'_j} + e^{(2\pi z'_j/q)\mathrm{i}}$
end for
$\{B_q(K_{w_i})\}_{i=0}^{L_{\mathrm{FFT}}-1} \leftarrow \mathrm{FFT}(Z)$, where $w_i = iq/L_{\mathrm{FFT}}$.
Find the value $i$ such that $|B_q(K_{w_i})|$ is maximal.
Output most significant $\log L_{\mathrm{FFT}}$ bits of $w_i$.
```

一次只能解 $l = l_{\mathrm{FFT}}$ 个密钥比特，需要多做几次。

设sk_{Hi}是由sk的l个高位比特组成的数，且sk有 λ 个比特。则：

$$
\begin{aligned}
k_i &\equiv z_i + h_i \cdot sk \mod n \\
&\equiv z_i + h_i \left(sk_{\mathrm{Hi}} \cdot 2^{\lambda-\ell} + sk_{\mathrm{Lo}} \right) \mod n \\
&\equiv z_i + h_i \cdot sk_{\mathrm{Hi}} \cdot 2^{\lambda-\ell} + h_i \cdot sk_{\mathrm{Lo}} \mod n.
\end{aligned}
$$

定义新的 $z_i := z_i + h_i \cdot sk_{\mathrm{Hi}} \cdot 2^{\lambda-l}$
用算法3继续算，直到全部算出为止。

Bleichenbacher细节及优化

RANGE REDUCTION和理论分析

Bleichenbacher优化

优化目标：数据量、空间和时间的权衡(tradeoff)，根据实际情况来优先照顾某一方面（例如签名数量不够就着重于减小数据量）

作者的改进：

- 正确地使用HGJ算法，一种range reduction时使用的算法
- 根据前人对研究GBP时对HGJ算法的分析，推广了一个权衡公式
- 使用线性规划来优化特定的方面

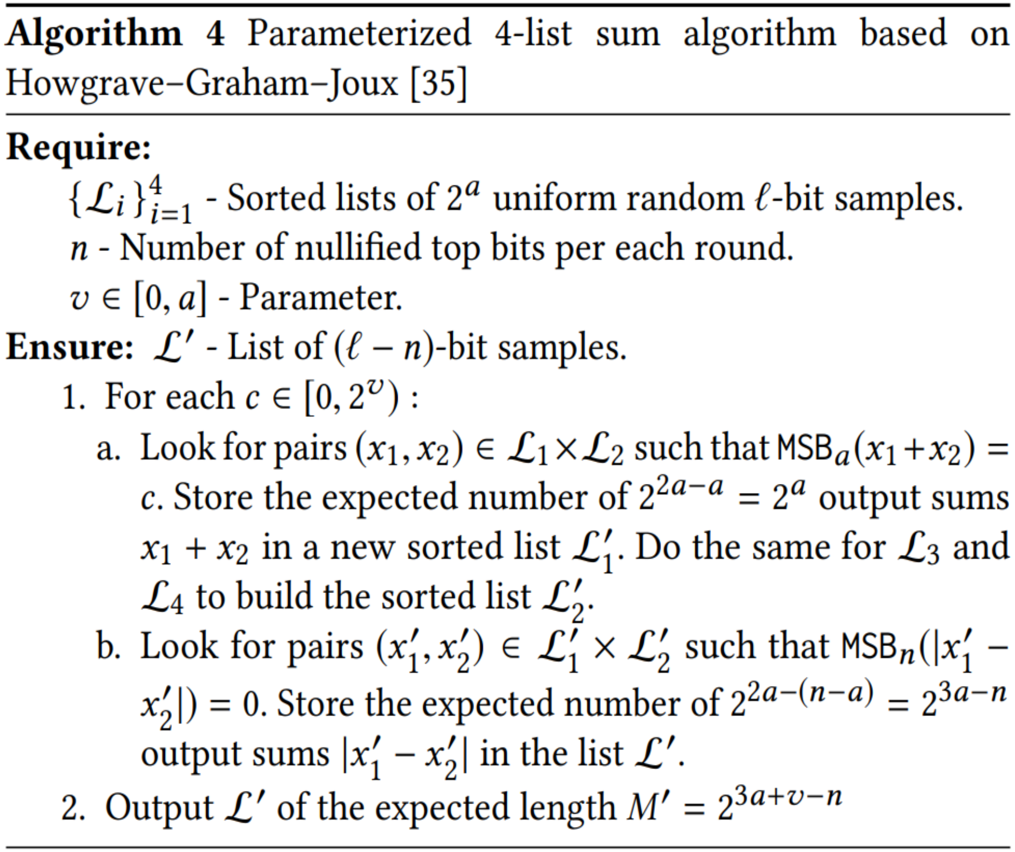
Range reduction & K-list sum

1. Sort-and-difference：排序，相邻相减
2. 基于格的方法：类似用格求解背包问题
3. **K-list sum问题(GBP问题)求解：找到较多h_i高位比特的碰撞，并且不一定是两两碰撞而是多个一起碰撞。**

（1和3可以做多轮）

定义：(K-list Sum问题) 给定K个排好序的列表$\mathcal{L}_1, \dots, \mathcal{L}_K$，其中每个都包含$2^a$个均匀选取的随机$l$比特整数。K-list sum问题是找到一个由 $x' = \sum_{i=1}^{K} \omega_i x_i$ 组成的非空列表$\mathcal{L}'$，其中 $(x_1, \dots, x_K) \in \mathcal{L}_1 \times \cdots \times \mathcal{L}_K$，$(\omega_1, \dots, \omega_K) \in \{-1, 0, 1\}^K$ 且满足对某个选定的正整数 $n \le l$，x'的 n 个MSB均为0。

HGJ算法求解K-List Sum



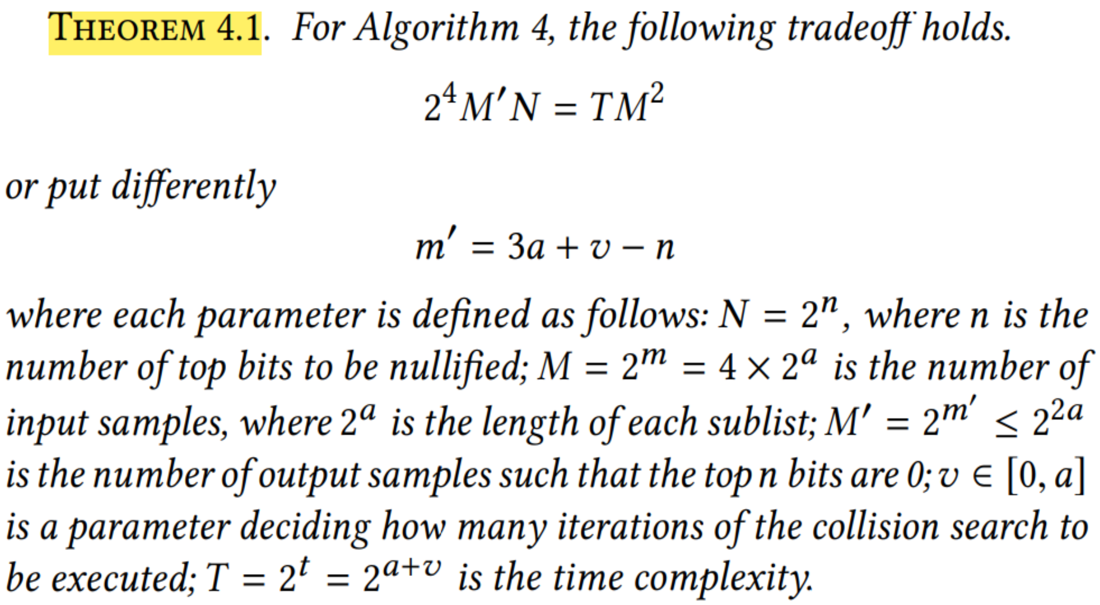
Algorithm 4 Parameterized 4-list sum algorithm based on Howgrave–Graham–Joux [35]

```
Require:
    $\{\mathcal{L}_i\}_{i=1}^4$ - Sorted lists of $2^a$ uniform random $\ell$-bit samples.
    $n$ - Number of nullified top bits per each round.
    $v \in [0, a]$ - Parameter.
Ensure: $\mathcal{L}'$ - List of $(\ell - n)$-bit samples.
  1. For each $c \in [0, 2^v)$ :
     a. Look for pairs $(x_1, x_2) \in \mathcal{L}_1 \times \mathcal{L}_2$ such that $\mathsf{MSB}_a(x_1 + x_2) = c$. Store the expected number of $2^{2a-a} = 2^a$ output sums $x_1 + x_2$ in a new sorted list $\mathcal{L}'_1$. Do the same for $\mathcal{L}_3$ and $\mathcal{L}_4$ to build the sorted list $\mathcal{L}'_2$.
     b. Look for pairs $(x'_1, x'_2) \in \mathcal{L}'_1 \times \mathcal{L}'_2$ such that $\mathsf{MSB}_n(|x'_1 - x'_2|) = 0$. Store the expected number of $2^{2a-(n-a)} = 2^{3a-n}$ output sums $|x'_1 - x'_2|$ in the list $\mathcal{L}'$.
  2. Output $\mathcal{L}'$ of the expected length $M' = 2^{3a+v-n}$
```

以$K = 4$为例，注意到：

1. 可以多次调用算法
2. 输出的(h'_i, z'_i)可以比输入的还要多，**数据量可以适当放大**。（先前的研究没有注意到这一点）

HGJ算法的理论分析



THEOREM 4.1. *For Algorithm 4, the following tradeoff holds.*

$$2^4 M' N = T M^2$$

or put differently

$$m' = 3a + v - n$$

where each parameter is defined as follows: $N = 2^n$*, where n is the number of top bits to be nullified;* $M = 2^m = 4 \times 2^a$ *is the number of input samples, where* 2^a *is the length of each sublist;* $M' = 2^{m'} \leq 2^{2a}$ *is the number of output samples such that the top n bits are 0;* $v \in [0, a]$ *is a parameter deciding how many iterations of the collision search to be executed;* $T = 2^t = 2^{a+v}$ *is the time complexity.*

Bleichenbacher优化目标：权衡攻击的**数据量**（签名数量）、**空间**、**时间**

目标：构造线性规划。

假设HGJ算法使用r轮，有r个等式：

$$m'_i = 3a_i + v_i - n_i,$$

$$(i = 0, \dots, r - 1)$$

并且有$m_{i+1} = m'_i$.

HGJ算法的理论分析

继续找等式和不等式。

- filtering technique：给定$2^{m_{\text{in}}}$个采样自均匀分布的l比特样本，对于任意的$f \geq 0$，可以只保留$2^{m_0} = 2^{m_{\text{in}}-f}$个样本的$l - f$个低位比特。
- small, sparse条件：

$$\log h_j' \leq \ell - f - \sum_{i=0}^{r-1} n_i \leq \ell_{\text{FFT}}$$

$$m_r = 2(\log\alpha - 4^r \log|B_q(\boldsymbol{K})|)$$

其中$\alpha \geq 1$是作者自己引入的新参数。可以调整 α 来控制bias函数的峰值与噪声值之间的差距。（论文中$\alpha = 8$）

数据量、空间和时间的权衡

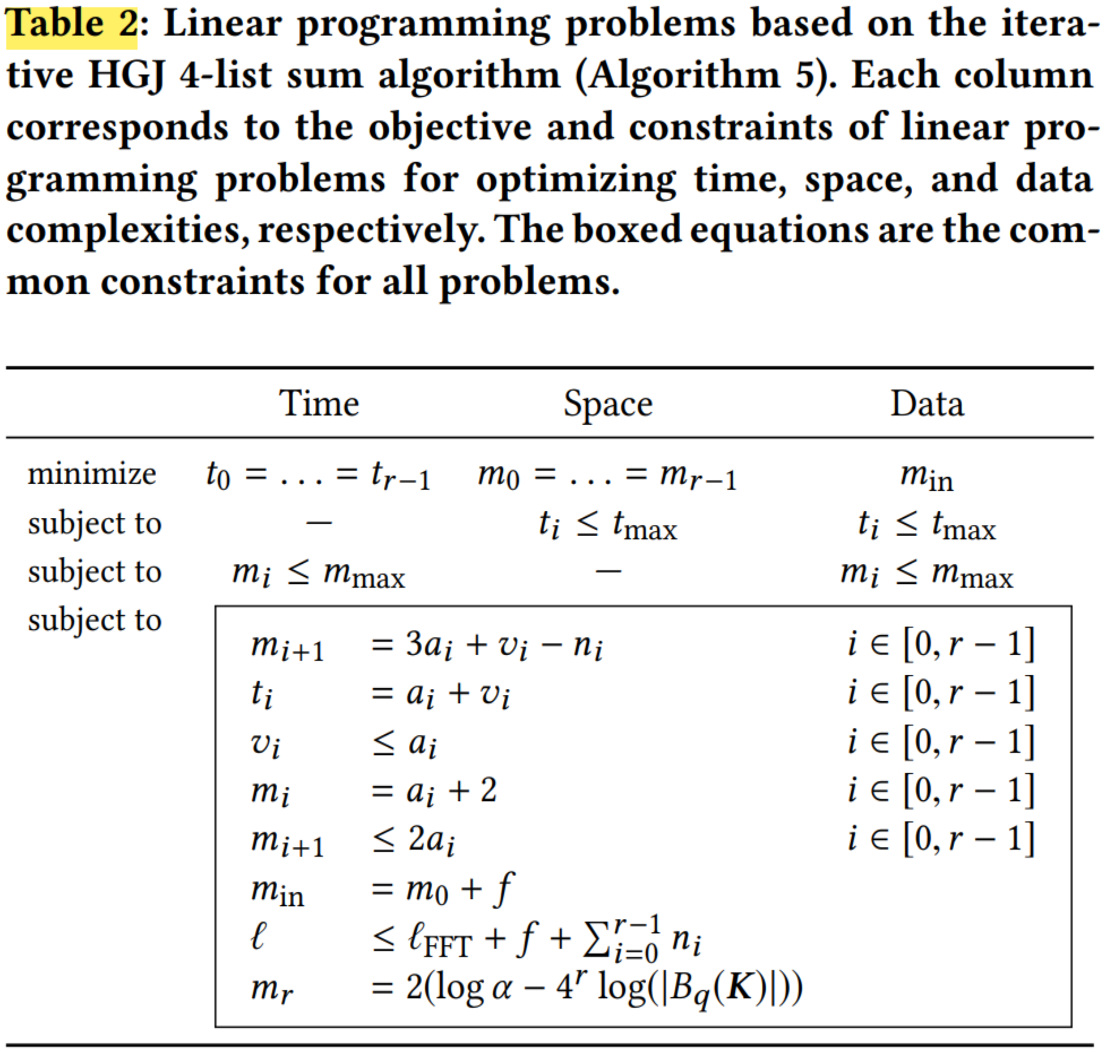


Table 2: Linear programming problems based on the iterative HGJ 4-list sum algorithm (Algorithm 5). Each column corresponds to the objective and constraints of linear programming problems for optimizing time, space, and data complexities, respectively. The boxed equations are the common constraints for all problems.

	Time	Space	Data
minimize	$t_0 = \ldots = t_{r-1}$	$m_0 = \ldots = m_{r-1}$	m_{in}
subject to	—	$t_i \leq t_{\max}$	$t_i \leq t_{\max}$
subject to	$m_i \leq m_{\max}$	—	$m_i \leq m_{\max}$
subject to	$\boxed{\begin{array}{lll} m_{i+1} & = 3a_i + v_i - n_i & i \in [0, r-1] \\ t_i & = a_i + v_i & i \in [0, r-1] \\ v_i & \leq a_i & i \in [0, r-1] \\ m_i & = a_i + 2 & i \in [0, r-1] \\ m_{i+1} & \leq 2a_i & i \in [0, r-1] \\ m_{\text{in}} & = m_0 + f & \\ \ell & \leq \ell_{\text{FFT}} + f + \sum_{i=0}^{r-1} n_i & \\ m_r & = 2(\log \alpha - 4^r \log(\lvert B_q(\boldsymbol{K}) \rvert)) & \end{array}}$		

数据量、空间和时间的权衡

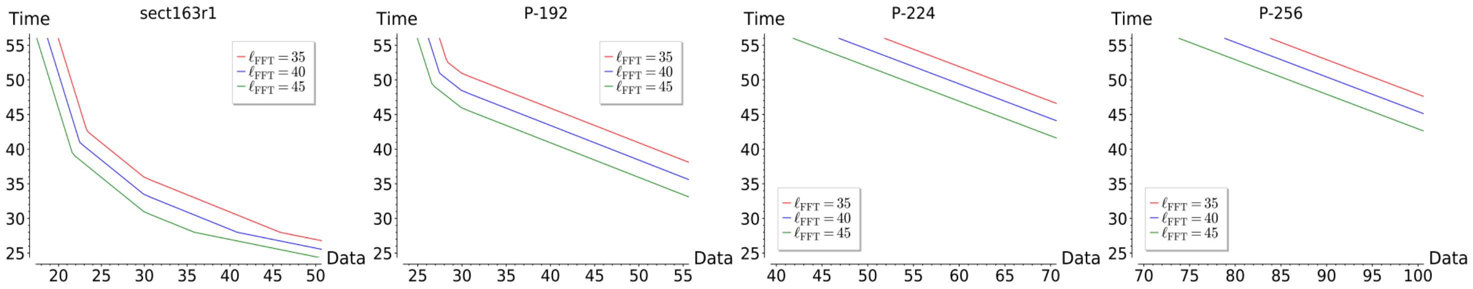


Figure 3: Time–Data tradeoffs where $m_{\max} = 30$, nonce k is 1-bit biased, slack parameter $\alpha = 8$ and the number of rounds $r = 2$.

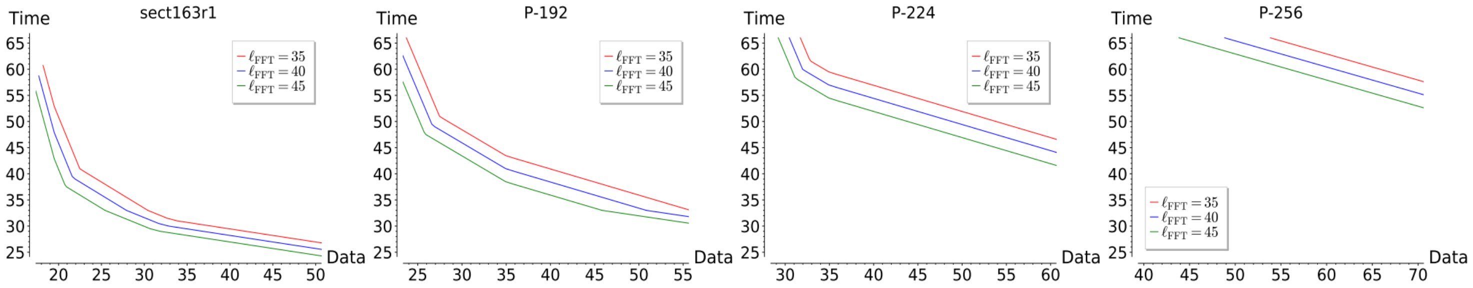


Figure 4: Time–Data tradeoffs where $m_{\max} = 35$, nonce k is 1-bit biased, slack parameter $\alpha = 8$ and the number of rounds $r = 2$.

- 多数情况下以优化数据量为目标
- 根据情况，时间为2^{50}甚至更高也可以接受

作者使用了96vCPU的亚马逊云计算集群，内存1TB

Bleichenbacher攻击的新记录

- 多数情况下以优化数据量为目标
- 根据情况，时间为2^{50}甚至2^{55}也可以接受

作者使用了24个96vCPU的AWS EC2，内存 4TB * 2 …

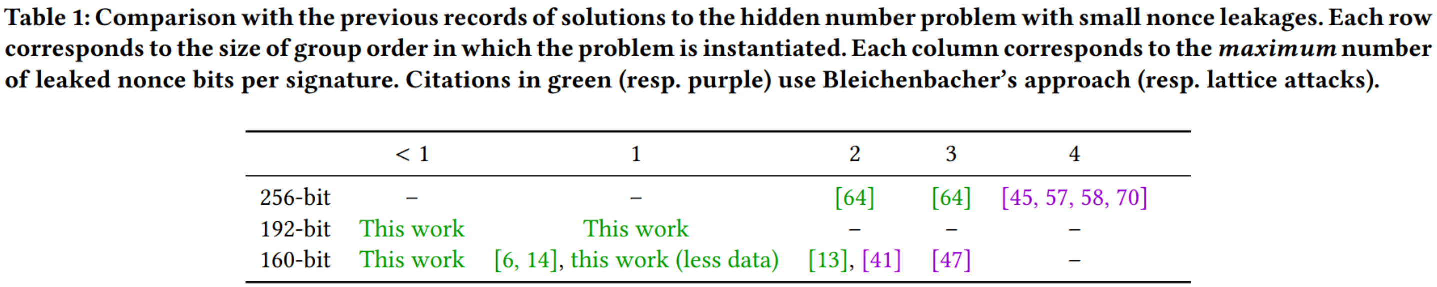


Table 1: Comparison with the previous records of solutions to the hidden number problem with small nonce leakages. Each row corresponds to the size of group order in which the problem is instantiated. Each column corresponds to the *maximum* number of leaked nonce bits per signature. Citations in green (resp. purple) use Bleichenbacher's approach (resp. lattice attacks).

	< 1	1	2	3	4
256-bit	–	–	[64]	[64]	[45, 57, 58, 70]
192-bit	This work	This work	–	–	–
160-bit	This work	[6, 14], this work (less data)	[13], [41]	[47]	–

总结

作者的贡献：

1. 发现侧信道漏洞LadderLeak，并利用漏洞获取nonce的一比特

2. 改进Bleichenbacher’s HNP attack的理论分析

3. 取得了新的HNP求解记录

对Bleichenbacher的优化适用于各种相关的侧信道漏洞，不局限于

Ladderleak这样的一比特泄露。

相关资料

1. LadderLeak: Breaking ECDSA With Less Than One Bit Of Nonce Leakage (CCS 2020)
2. New Bleichenbacher Records: Fault Attacks on qDSA Signatures (TCHES 2018)
3. Talk: Ladderleak (WAC 2020 - Workshop on Attacks in Cryptography)
4. Talk: LadderLeak (Blackhat Europe 2020)

THANK YOU

Q&A