LadderLeak: Breaking ECDSA With Less Than One Bit Of Nonce Leakage

Diego F. Aranha, Felipe Rodrigues Novaes, Akira Takahashi, Mehdi Tibouchi, Yuval Yarom

CCS 2020

Presentation by Srushti Singh & Jonathan Poveda Colominas

07.04.2025

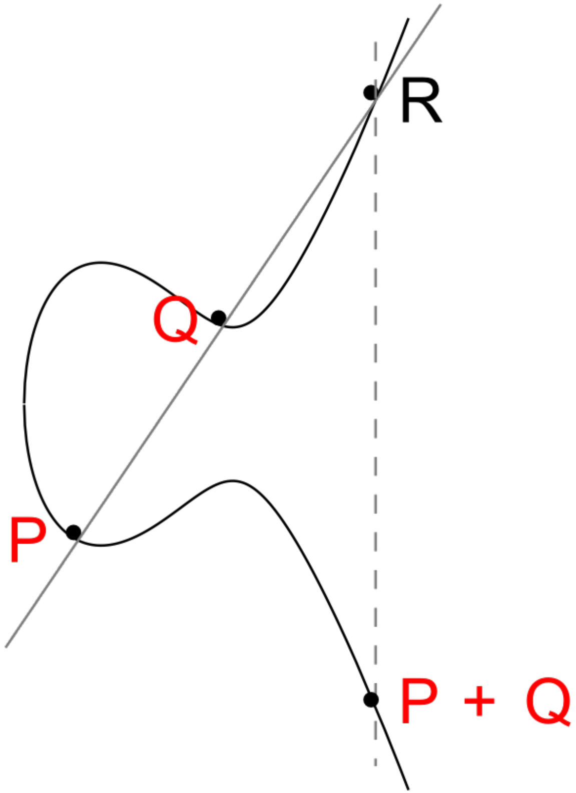
Contents

1. Background
2. *LadderLeak*
3. Hidden Number Problem (HNP) and ECDSA
4. Bias function
5. Bleichenblacher's attack framework
6. $\mathcal{K}$-list sum problem
7. Unified Time-Space-Data tradeoffs
8. Experiments and results

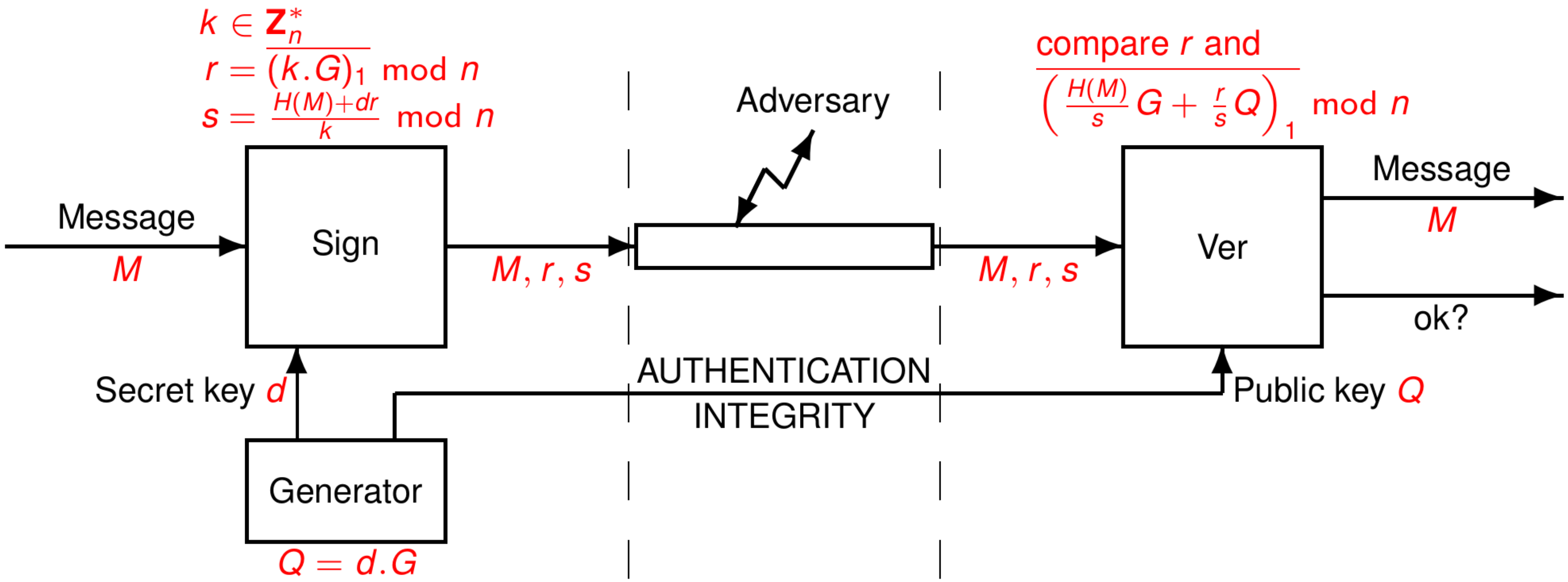
Background

Elliptic curve cryptography

- Curve defined over finite field $\mathbb{F}$
- With point at infinity O and group law $+$, defines a group
  - O identity element
- $P + Q$ found with “chord-and-tangent” rule
- Scalar multiplication $[k]P = P + P + \cdots + P$
- Finding k given $(P, [k]P)$ is hard on certain curves (ECDLP)



ECDSA



$k \in \mathbf{Z}_n^*$

$r = (k.G)_1 \bmod n$

$s = \frac{H(M)+dr}{k} \bmod n$

Adversary

compare r and

$\left(\frac{H(M)}{s} G + \frac{r}{s} Q\right)_1 \bmod n$

Message M

Sign

M, r, s

M, r, s

Ver

Message M

ok?

Secret key d

AUTHENTICATION

INTEGRITY

Public key Q

Generator

$Q = d.G$

select field, elliptic curve
G point of order n
n prime

Nonce is sensitive

- Nonce k should **NEVER** be leaked
- In case of full leakage, secret key sk easily recovered

$$s = \frac{H(m) + \mathrm{sk} \cdot r}{k} \Leftrightarrow \mathrm{sk} = \frac{k \cdot s - H(m)}{r}$$

LadderLeak

Overview

- **Less than** one bit of leakage ?
  - Uses the leak of one most significant bit (MSB) of the nonce
  - Leak can have a probability < 1 to be correct
  - Needs MSBs from many different nonces
- Combines methods based on the Hidden Number Problem (HNP) and Discrete Fourier Transform (DFT)
- Can be tweaked depending on available resources and needs
- Leakage is part of the paper

Hidden Number Problem (HNP) and ECDSA

HNP with erroneous input

- q a prime number
- $\mathrm{sk} \in \mathbb{Z}_q$ a secret.
- $h_i, k_i \in \mathbb{Z}_q$ uniformly distributed for all $i = \{1, ..., M\}$
- $z_i = k_i - h_i \cdot \mathrm{sk} \bmod q$
- χ_b a distribution on $\{0,1\}^b$ for $b > 0$
- $\mathrm{EMSB}_{\chi_b}(x) = \mathrm{MSB}_b(x) \oplus e$
  - e is a b bits error string sampled from χ_b

HNP with erroneous input asks one to find sk given M samples (h_i, z_i) and $\mathrm{EMSB}_{\chi_b}(k_i)$

ECDSA as the HNP

ECDSA signatures from the same private key with leaky nonces are instances of the same HNP

$$s_i = \frac{H(m_i) + \mathrm{sk} \cdot r_i}{k_i} \Leftrightarrow \underbrace{\frac{H(m_i)}{s_i}}_{z_i} = k_i - \underbrace{\frac{r_i}{s_i}}_{h_i} \mathrm{sk}$$

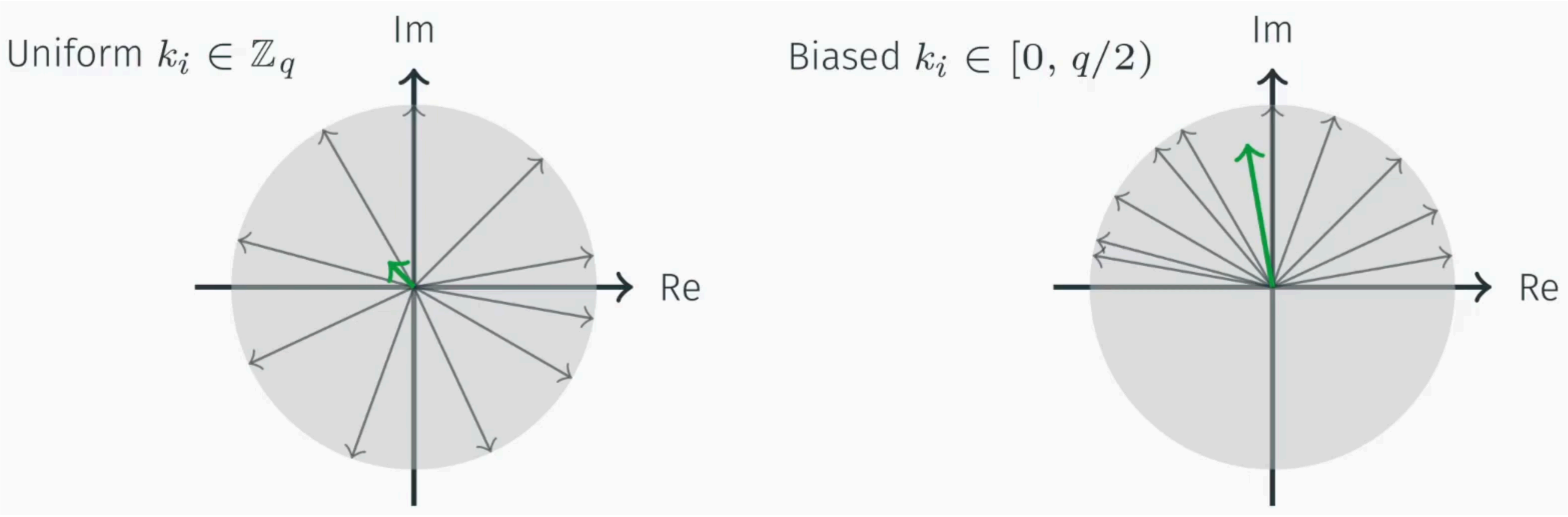
Bias function

Bias function

- Goal: Function B_q quantifies (modular) bias of a collection of samples $K = \{k_i\}_{i=1}^{M}$
  - $B_q(K) \approx 1$ when K is a collection of biased samples in $\mathbb{Z}_q$
  - $B_q(K) \approx 0$ when K is a collection of uniformly distributed samples in $\mathbb{Z}_q$
- Idea: Use the inverse discrete Fourier transform (DFT)

$$B_q(K) = \frac{1}{M} \sum_{i=1}^{M} e^{2\pi i \frac{k_i}{q}}$$

Intuition



$$B_q(K) = \frac{1}{M} \sum_{i=1}^{M} e^{2\pi i \frac{k_i}{q}}$$

Bias function

For l fixed MSBs of the samples, the bias function's magnitude can be approximated (for a large q) with

$$|B_{q(K)}| \approx \frac{2^l}{\pi} \cdot \sin\left(\frac{\pi}{2^l}\right)$$

For $l = 1$, $|B_{q(K)}| \approx 0.637$

Erroneous form

Attack focuses on inputs with an error on the MSB of $\varepsilon \in \left[0, \frac{1}{2}\right]$, bias should take this into account.

Let $b \in \{0, 1\}, \varepsilon \in \left[0, \frac{1}{2}\right]$ and an even integer $q > 0$. Let $\boldsymbol{K}$ be a random variable with the following distribution over $\mathbb{Z}_q$

$$\Pr(\mathrm{MSB}(\boldsymbol{K}) = 0) = (1 - b)\frac{1 - \varepsilon}{q/2} + b\frac{\varepsilon}{q/2}$$

$$\Pr(\mathrm{MSB}(\boldsymbol{K}) = 1) = b\frac{1 - \varepsilon}{q/2} + (1 - b)\frac{\varepsilon}{q/2}$$

The modular bias of $\boldsymbol{K}$ is

$$B_q(\boldsymbol{K}) = (1 - 2\varepsilon)B_q(\boldsymbol{K}_b)$$

for $\boldsymbol{K}_b$ uniformly distributed over $\left[b\frac{q}{2}, (b + 1)\frac{q}{2}\right]$

Bleichenblacher's attack framework

Idea to solve HNP and naive approach

- $B_q(K)$ quantifies bias on $K = \{k_i\}_{i=1}^M = \{z_i + h_i \cdot \mathrm{sk}\}_{i=1}^M$
- HNP states that we know $\{(z_i, h_i)\}_{i=1}^M$
- We can compute $K_w = \{z_i + h_i \cdot w\}_{i=1}^M$ for any $w \in \mathbb{Z}_q$
- Observation: $|B_q(K_w)|$ is highest when $w = \mathrm{sk}$
  - $|B_q(K_w)| \approx \frac{1}{\sqrt{M}}$ when $w \neq \mathrm{sk}$
- We could try all possible $w \in \mathbb{Z}_q$
  - Not better than exhaustive search

Collision search

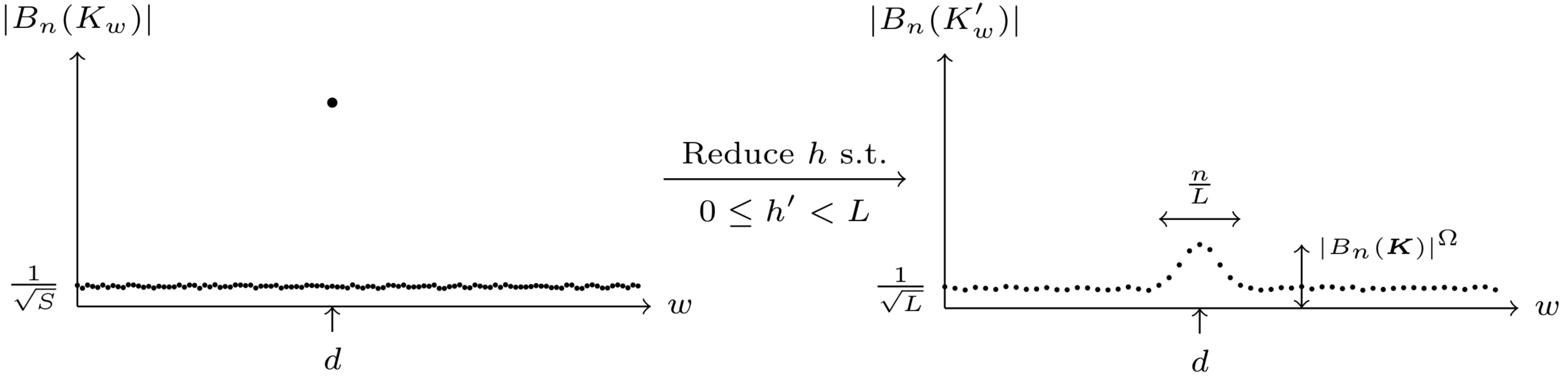
- Observation: Linear combinations generating M' new samples $\left\{\left(h'_j, z'_j\right)\right\}_{j=1}^{M'}$ where $h'_j < L_{\mathrm{FFT}}$ broaden the peak's width to approximately $\frac{q}{L_{\mathrm{FFT}}}$
- Reduces number of candidate points to L_{FFT}
  - Can be tweaked taking into account $O(L_{\mathrm{FFT}} \log L_{\mathrm{FFT}})$ time and $O(L_{\mathrm{FFT}})$ space
- Downside: Peak height reduces exponentially with the number of linear combinations

For coefficients $\omega_{i,j} \in \{-1, 0, 1\}$ s.t

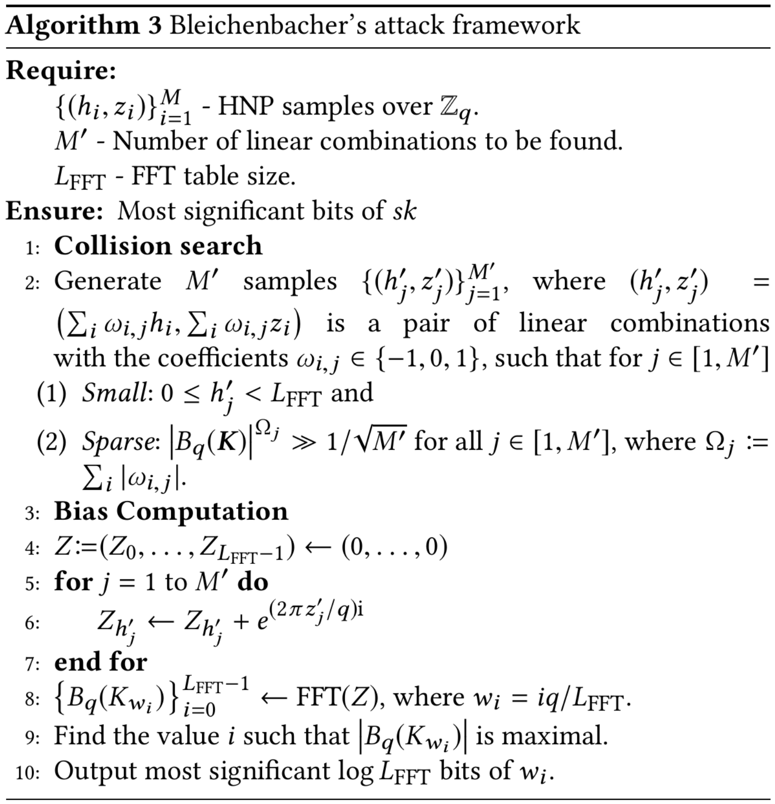
$$\left\{\left(h'_j, z'_j\right)\right\}_{j=1}^{M'} = \left\{\left(\sum_i \omega_{i,j} h_i, \sum_i \omega_{i,j} z'_i\right)\right\}_{i=1}^{M'}$$

The new peak height is $|B_{q(K)}|^{\Omega_j}$ where $\Omega_j = \sum_i |\omega_{i,j}|$

Collision search



The attack



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

Note: HNP samples are chosen according to the MSB of the leaked nonce

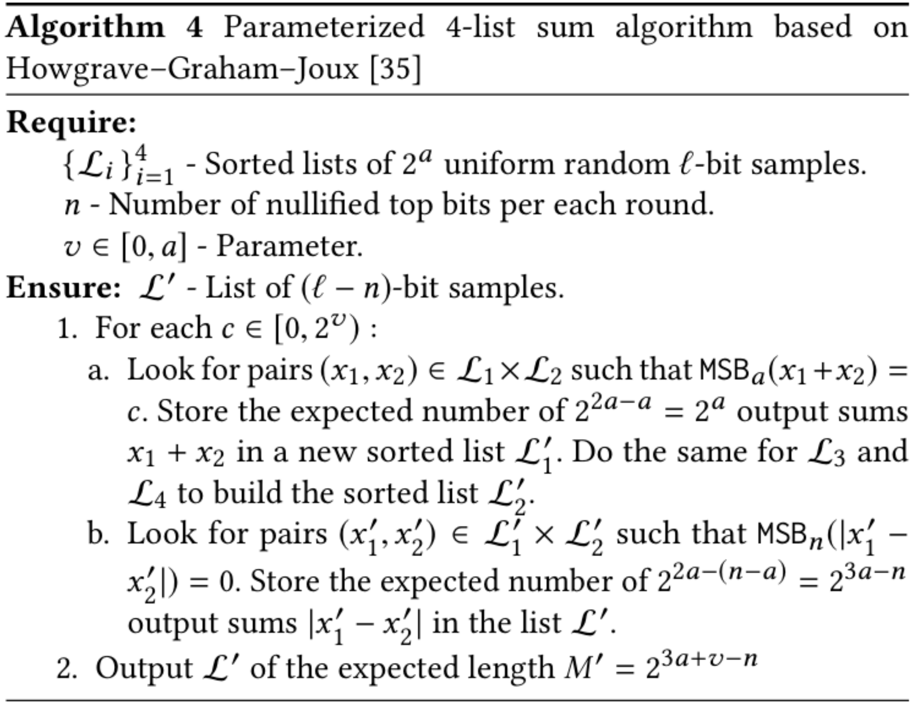
$\mathcal{K}$-LIST SUM PROBLEM

$\mathcal{K}$-list sum problem

Definition

Given $\mathcal{K}$ sorted lists $L_1, ..., L_K$, each of which consists of 2^a uniformly random l-bit integers, the $\mathcal{K}$-list sum problem consists of finding a non-empty list L' consisting of $x' = \sum_{i=1}^{K} \omega_i x_i$, where $\mathcal{K}$-tuples $(x_1, ..., x_K) \in L_1 \times ... \times L_K$ and $(\omega_1, ..., \omega_K) \in \{-1, 0, 1\}^K$ satisfy $\mathrm{MSB}_n(x') = 0$ for some target parameter $n \leq l$

4-list sum problem

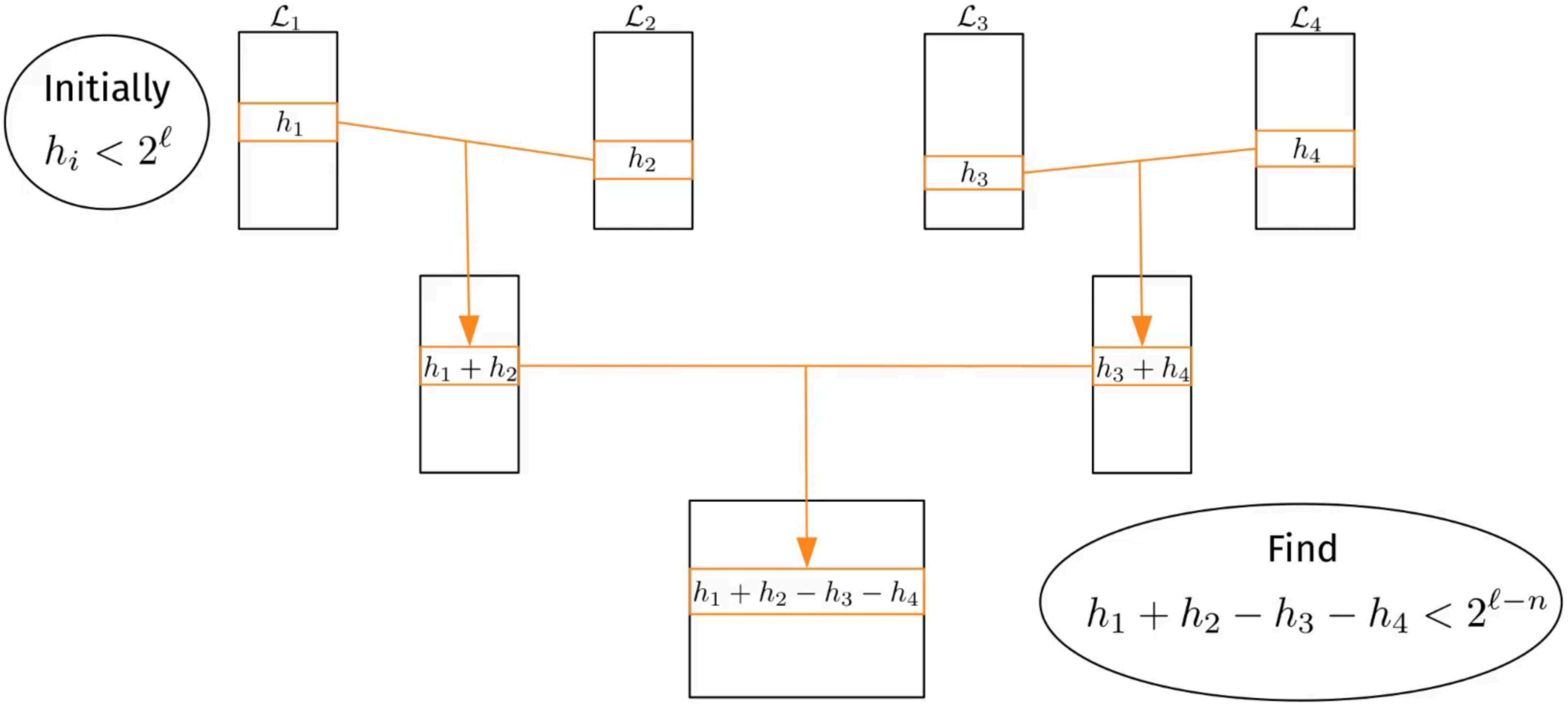


Algorithm 4 Parameterized 4-list sum algorithm based on Howgrave–Graham–Joux [35]

```
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 $M' = 2^{3a+v-n}$
```

- The 4-list sum algorithm efficiently reduces candidate nonces by performing a collision search to generate M' samples with the top n_i bits null while respecting the small and sparse linear combination properties.

4-list sum problem



Unified Time-Space-Data tradeoffs

Tradeoffs for parametrized 4-list Sum algorithm

- Time-space tradeoffs analyzed in the previous works such as HGJ approach made two artificial assumptions:

1. $M = M' = \mathrm{L}_{\mathrm{FFT}}$ in the Bleichenbacher's attack framework.
2. The number of collided bits is a fixed constant in the 4-list sum algorithm

- a third parameter of data complexity expanded to the time-space complexity tradeoffs

=> a "mild" generalization of Dinur's tradeoff formula for our parametrized 4-list sum algorithm

Tradeoffs for parametrized 4-list Sum algorithm

- Following tradeoff holds for the 4-list sum problem approached in the paper:

$$2^4 M' N = \mathrm{TM}^2 \Leftrightarrow m' = 3a + v - n$$

- $N = 2^n$ => n := # of top bits to be nullified
- $M = 2^m = 4 \times 2^a$ => # of input samples
- 2^a => length of each sublist
- $M' = 2^{m'} \leq 2^{2\mathrm{a}}$ => # of output samples s.t. top n bits are 0
- $v \in [0, a]$ => parameter deciding # of iterations of collision search
- $T = 2^t = 2^{\mathrm{a}\ +\mathrm{v}}$ => time complexity.
- Gives more flexibility to sample amplification

Integration with Bleichenbacher and Linear Programming

- Construct a linear programming problem by integrating:
  1. tradeoff formula for 4-list sum algorithm &
     - Following two constraints of Bleichenbacher's attack framework
  2. small linear combination
  3. sparse linear combination

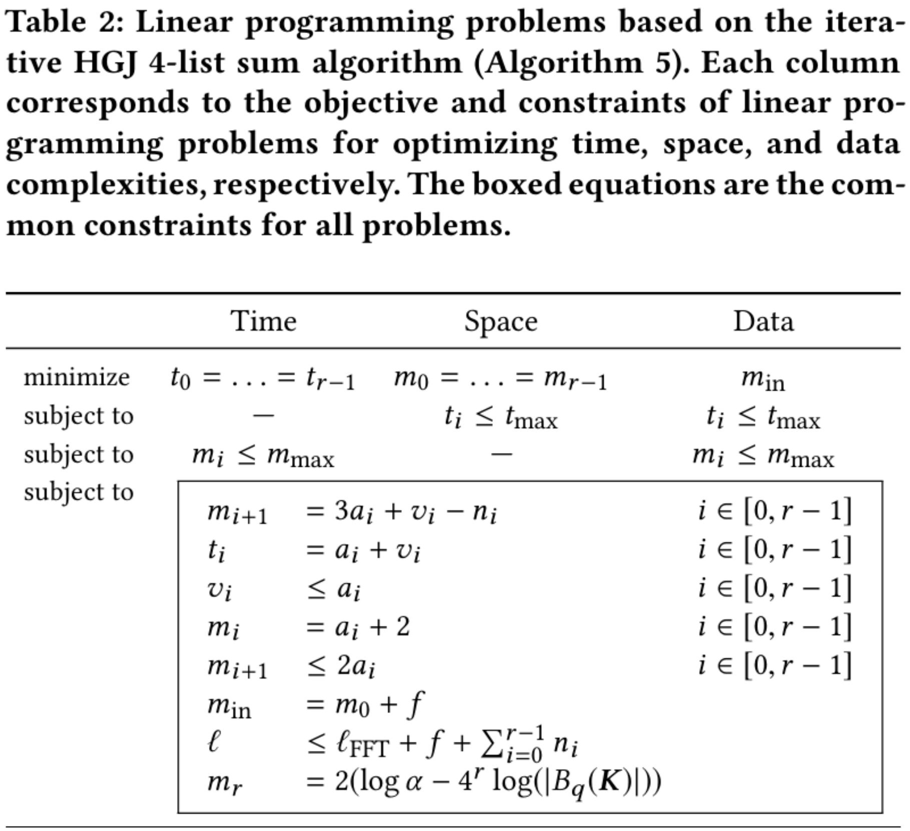


Table 2: Linear programming problems based on the iterative HGJ 4-list sum algorithm (Algorithm 5). Each column corresponds to the objective and constraints of linear programming problems for optimizing time, space, and data complexities, respectively. The boxed equations are the common constraints for 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	$m_{i+1} = 3a_i + v_i - n_i$		$i \in [0, r-1]$
	$t_i = a_i + v_i$		$i \in [0, r-1]$
	$v_i \leq a_i$		$i \in [0, r-1]$
	$m_i = a_i + 2$		$i \in [0, r-1]$
	$m_{i+1} \leq 2a_i$		$i \in [0, r-1]$
	$m_{\text{in}} = m_0 + f$		
	$\ell \leq \ell_{\text{FFT}} + f + \sum_{i=0}^{r-1} n_i$		
	$m_r = 2(\log \alpha - 4^r \log(\lvert B_q(\boldsymbol{K})\rvert))$		

Integration with Bleichenbacher and Linear Programming

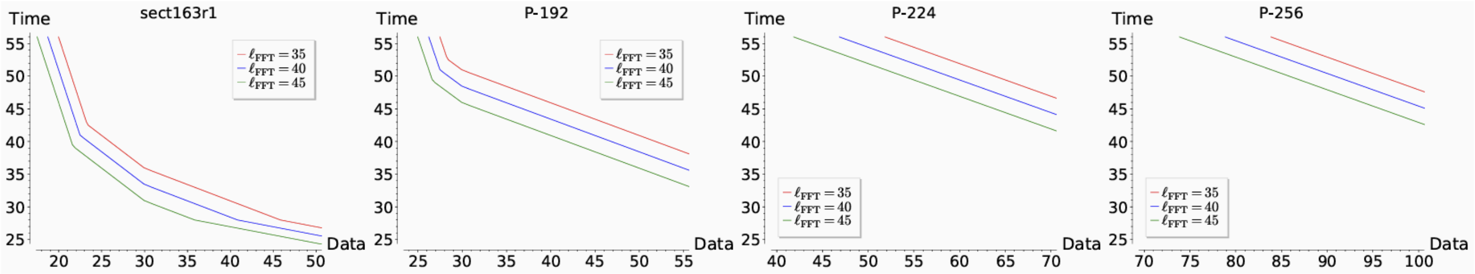


Figure 3: Time–Data tradeoffs where $m_{\text{max}} = 30$, nonce k is 1-bit biased, slack parameter $\alpha = 8$ and the number of rounds $r = 2$.

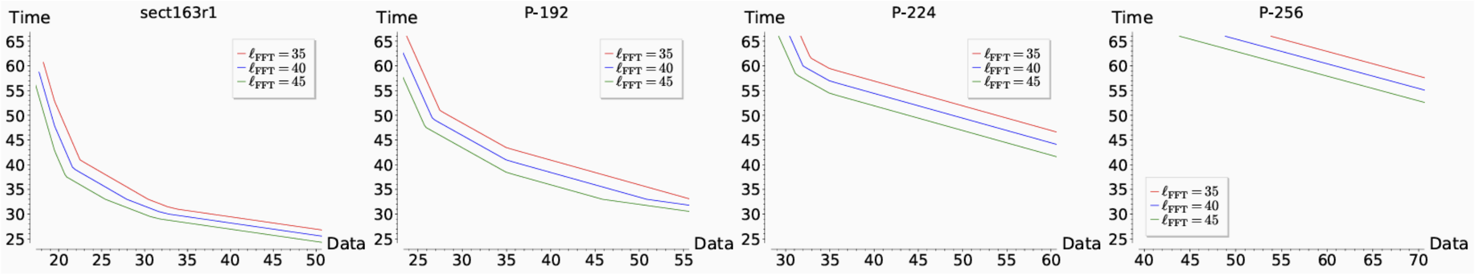


Figure 4: Time–Data tradeoffs where $m_{\text{max}} = 35$, nonce k is 1-bit biased, slack parameter $\alpha = 8$ and the number of rounds $r = 2$.

- optimal time and data complexities for attacking a 1-bit biased HNP with varying FFT sizes and maximum memory bounds

Experiments and results

Experiments and results

- Attack successfully implemented
- Used modified version of OpenSSL 1.0.2u
  - Modifications for convenience in MSB leakage
- Recovered secret keys on P-192 and sect163r1 curves

Experiments and results

- P-192 (~96 bits security)
  - 24 AWS instances with 96 vCPUs for collision search
  - 2 AWS instances with 4 TB of RAM for FFT tables of size $L_{\mathrm{FFT}} = 2^{38}$ entries of 32B

ε	Input	Output	L_{FFT}	Total time	Recovered MSBs
0	2^{29}	2^{27}	2^{38}	113.5h	39
0.01	2^{35}	2^{30}	2^{37}	64h	39

Experiments and results

- sect163r1 (~80 bits security)
  - Cluster of 16 nodes with 16 core CPUs + 128 GB RAM for $\varepsilon = 0$
  - 48-core + 512 GB RAM workstation for $\varepsilon = 0.027$

ε	Input	Output	L_{FFT}	Total time	Recovered MSBs
0	2^{23}	2^{27}	2^{35}	8h	36
0.027	2^{24}	2^{29}	2^{34}	43h	35

Sources

- https://www.maths.ox.ac.uk/system/files/media/picture3.png
- Cryptography and security (COM-401), fall 2024
- Presentation of LadderLeak https://youtu.be/UbjOKMTVMWQ
- Takahashi, Akira, Mehdi Tibouchi, and Masayuki Abe. “New Bleichenbacher records: Fault attacks on qDSA signatures.” Cryptology ePrint Archive (2018).