Secret Key Recovery from Partial Information in the Pre- and Post-Quantum World

Alex May

Ruhr-University Bochum
CASA - casa.rub.de/en/

SUMMER SCHOOL ON REAL-WORLD CRYPTO AND PRIVACY@VODICE 2023

What's the task?

Setting: Public key world with key pairs (pk, sk), $c = \mathrm{Enc}_{pk}(m)$ is an encrypted message.

A cryptanalyst's job

1. **Secret Key Recovery:** Given *pk*, reconstruct *sk*.
2. **Message Recovery :** Given *pk* and *c*, reconstruct *m*.

Running examples:

1. RSA: Given $(N = pq, e), c = m^e \bmod N$, recover either $p, q, d = e^{-1} \bmod \phi(N)$, or m.
2. ElGamal: Given $(g, g^a), c = (g^b, g^{ab}m)$, recover either a (dlog), or m (DH).
3. McEliece: Let's wait a bit.
4. Kyber, Dilithium, Falcon (LWE), NTRU: Given $A \in \mathbb{Z}_q^{n \times n}, \mathbf{b} = A\mathbf{s} + \mathbf{e} \bmod q$, recover $\mathbf{s}$.

Question: What if too hard?

"Factoring is hard. Let's go shopping!" (Nadia Heninger)
"Dlog is hard. Let's go swimming!" (Pierrick Gaudry in Vodice, 2 days ago)

Let us do a little bit of cheating.

Side-Channel Attacks

Try to get some useful information about sk or m from side-channels.

Various side-channel sources:

1. Power Consumption
2. Timing
3. Faults
4. Cold Boot
5. Micro Architectural

New, easier task

Given pk and some information about sk or m, recover the latter (in polynomial time).

Maybe a more rewarding task.

Partial Key Exposure Attack, or better: Partial Key Completion

1. **Secret Key Recovery :** Given pk + information on sk, reconstruct sk.
2. **Message Recovery :** Given pk and c + information on m, reconstruct m.

Information can be:

- some bits in consecutive positions
- some bits in random positions
- all bits with some error probability
- many other things

Leaky Intuition.

If sk (or m) has n bits, we leak k bits, then the problem should retain $n - k$ bit hardness.
Fascinating (at least for me): This intuition is often plain wrong!

Factoring with High Bits Known

Theorem (Coppersmith 1996)

Let $N = pq$ be an RSA modulus, p can be recovered with $1/2$ of its most significant bits.

- Model as $N = (\tilde{p} + x)y$.
- Remaining bits are root of $f(x, y) = N - (\tilde{p} + x)y$.
- Coppersmith method: Finds all roots with $|xy| \leq N^{\frac{3}{4}}$.

RSA Partial Key Exposure Results on Secret Exponent d

Small secret d: $ed = 1 \bmod \phi(N)$

- Modelling as $ed = 1 + k(N - (p + q - 1))$
- Wiener (1990): $f(x, y) = ex - y \bmod N$, works for $d \leq N^{0.25}$.
- Boneh-Durfee (1999): $f(x, y) = 1 + x(N - y) \bmod e$, works for $d \leq N^{0.29}$.

RSA Partial Key Exposure: d not small, but known bits

- Ernst, Jochemsz, May, de Weger (2005): Known bits extension of Boneh-Durfee.
- Takayasu, Kunihiro (2014): Several nice improvements.

Suggestion (don't take too serious): Make wild conjectures.

Boneh-Durfee conjectured $d \leq N^{0.5}$. Stimulated lots of research, although likely not true.

RSA Partial Key Exposure Results on Secret CRT Exponents

Small secret d_p, d_q**:** $ed_p = 1 \bmod p-1$ and $ed_q = 1 \bmod q-1$

- Jochemsz, May (2007): Attack for $d_p, d_q \le N^{0.073}$.
- Takayasu, Lu, Peng (2017): Attack for $d_p, d_q \le N^{0.122}$.

RSA Partial Key Exposure: d_p, d_q not small, but known bits

- May, Nowakowski, Sarkar (2021): Extension of Takayasu, Lu, Peng.
- May, Nowakowski, Sarkar (2022): 1/3 bits of d_p, d_q suffice for $e \approx N^{\frac{1}{12}}$.

Some surveys (only those written by myself):

1. New RSA Vulnerabilities Using Lattice Reduction Methods (2003)
2. Using LLL-Reduction for Solving RSA and Factorization Problems: A Survey (2007)
3. Lattice-based Integer Factorization - An Introduction to Coppersmith's Method (2021)

Personal remark

I do not want to do this anymore (I swear, really), but it haunts me!

Discrete Logarithms

Discrete Logarithm (dlog): public key (g, g^a), let g be of order q (n bits)

- Pollard Rho (1975): $\mathcal{O}(\sqrt{q})$ steps
- Pollard Lambda (1975): small $a \leq 2^k \ll q$, works in $\mathcal{O}(\sqrt{2^k})$.

Partial Key Exposure:

- Pollard Lambda (1975): Given $n-k$ upper bits of a, recover a in $\mathcal{O}(\sqrt{2^k})$.

$$\frac{g^a}{g^{\tilde{a}}} = g^{a-\tilde{a}}$$

- Esser, May (2020): Low weight dlog problem from bits in random positions of a.

Conclusion

Almost no results, nothing polynomial! Why the heck, actually?

Diffie-Hellman Problem

DH problem: Given g^a, g^b, compute g^{ab}.

- Boneh, Venkatesan (1996): Hidden Number Problem
- Provides algorithm that computes g^{ab} from an algorithm for MSBs of g^{ab}.

Partial Key Exposure

- Successfully applied for ECDSA with small bit leakage per signature, lots of research, see e.g. Albrecht, Heninger (2021) for a recent one.

Once again

Dlog still seems to be more resistant than RSA. Is Partial Key an RSA artefact?

What about the Post-Quantum World?

Common belief

Modern Post-Quantum systems are not really vulnerable to Partial Key Exposure.

Some results:

1. Albrecht, Deo, Paterson
   "Cold boot attacks on Ring and Module LWE keys under the NTT" (2018)
2. Dachman-Soled, Ducas, Gong, Rossi
   "LWE with Side Information: Attacks and Concrete Security Estimation" (2020)
3. Esser, May, Verbel, Wen
   "Partial Key Exposure Attacks on BIKE, Rainbow and NTRU" (2022)
4. Kirshanova, May
   "Decoding McEliece with a Hint-Secret Goppa Key Parts Reveal Everything" (2022)

The (Classic) McEliece Cryptosystem (1978, 2022)

Security: Based on hardness of decoding binary linear codes.

Definition Linear Code

A binary linear code C is a k-dimensional subspace of $\mathbb{F}_2^n$.

- Via generator matrix $G \in \mathbb{F}_2^{k \times n}$:

$$C = \{\mathbf{x}G \mid \mathbf{x} \in \mathbb{F}_2^k\}.$$

- Via parity check matrix $H \in \mathbb{F}_2^{(n-k) \times n}$:

$$C = \{\mathbf{c} \in \mathbb{F}_2^n \mid H\mathbf{c} = \mathbf{0}\}.$$

Classic McEliece:

- sk is a structured parity check matrix H of a Goppa code.
- pk is (randomly scrambled) version of H.
- Smallest parameters: $n = 3488$, $n - k = 768 = 64 \cdot 12 = tm$.

The McEliece Secret Key

Setting: We work with

- a "large" field $\mathbb{F}_{2^m} = \mathbb{F}_{2^{12}}$,
- a list of $n = 3488$ Goppa points $L = (\alpha_1, \ldots, \alpha_n) \in \mathbb{F}_{2^m}^n$,
- an irreducible $\deg$-t ($\deg$-64) Goppa polynomial $g(x) \in \mathbb{F}_{2^m}[x]$.

Definition Goppa code / McEliece secret key

We define a Goppa code as

$$C(L, g) = \left\{ \mathbf{c} \in \mathbb{F}_2^n \ : \ \sum_{i=1}^{n} \frac{c_i}{x - \alpha_i} \equiv 0 \mod g(x) \right\}.$$

A McEliece secret key is $sk = (L, g)$.

Construction of Public Key

McEliece keys:

- Parity check matrix $\overline{H}(L,g) \in \mathbb{F}_{2^m}^{t \times n}$ for $C(L,g)$:

$$\overline{H}(L,g) = \begin{pmatrix} 1 & 1 & \dots & 1 \\ \alpha_1 & \alpha_2 & \dots & \alpha_n \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{t-1} & \alpha_2^{t-1} & \dots & \alpha_n^{t-1} \end{pmatrix} \cdot \begin{pmatrix} g^{-1}(\alpha_1) & 0 & \dots & 0 \\ 0 & g^{-1}(\alpha_2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & g^{-1}(\alpha_n) \end{pmatrix}.$$

- Mapping $\mathbb{F}_{2^m} \to \mathbb{F}_2^m$ yields parity check matrix in $\mathbb{F}_2^{tm \times n}$.
- Partial Gaussian elimination of this matrix gives *McEliece public key* $pk = H \in \mathbb{F}_2^{tm \times n}$.

First Partial Key Exposure

Theorem Folklore Result

On input $pk = H \in \mathbb{F}_2^{tm \times n}$ and $L = (\alpha_1, \ldots, \alpha_n)$, one can recover $g(x)$ in polynomial time.

Idea:

- Compute codeword $\mathbf{c} = (c_1, \ldots, c_n) \in \mathbb{F}_2^n \setminus \mathbf{0}$ with $H\mathbf{c} = \mathbf{0}$.
- Since $\mathbf{c} \in C(L, g)$ we have

$$\sum_{i=1}^{n} \frac{c_i}{x - \alpha_i} \equiv 0 \mod g(x).$$

- Multiplication by $\prod_{j=1,\ldots,n}(x - \alpha_j)$ yields

$$\sum_{i=1}^{n} c_i \prod_{1 \le j \le n, j \ne i} (x - \alpha_j) \equiv 0 \bmod g(x).$$

- Factor left hand side, and look for degree-t irreducible $g(x)$.

Using only $tm + 1$ Goppa Points.

Theorem Kirshanova, May (2022)

On input $pk = H \in \mathbb{F}_2^{tm \times n}$ and $(\alpha_i)_{i \in \mathcal{I}}$, $|\mathcal{I}| = tm + 1$, one recovers $g(x)$ in polynomial time.

Idea:

- Let $H' \in \mathbb{F}_2^{tm \times (tm+1)}$ be the restriction on H on columns in $\mathcal{I}$.
- Compute $\mathbf{c}' \neq \mathbf{0}$ with $H'\mathbf{c}' = \mathbf{0}$. Expand $\mathbf{c}'$ with 0's outside $\mathcal{I}$ to $\mathbf{c}$.
- We obtain codeword $\mathbf{c} = (c_1, \ldots, c_n) \in \mathbb{F}_2^n$ with $H\mathbf{c} = \mathbf{0}$, $\mathrm{supp}(\mathbf{c}) \in \mathcal{I}$.
- Since $\mathbf{c} \in C(L, g)$ we have

$$\sum_{i \in \mathcal{I}} \frac{c_i}{x - \alpha_i} \equiv 0 \mod g(x).$$

- Multiplication by $\prod_{j \in \mathcal{I}} (x - \alpha_j)$ yields

$$\sum_{i \in \mathcal{I}} c_i \prod_{j \in \mathcal{I} \setminus \{i\}} (x - \alpha_j) \equiv 0 \bmod g(x).$$

- Factor left hand side, and look for degree-t irreducible $g(x)$.

Experimental Results.

Observation

Improvement comes from capability of computing codewords of weight at most $tm+1$.

(n,t,m)	$\ell = tm+1$	$\lvert\mathcal{L}\rvert = 1$	$\ell = tm+2$	$\lvert\mathcal{L}\rvert = 1$	Av. time
$(3488, 64, 12)$	769	97%	770	100%	18 sec
$(4608, 96, 13)$	1249	99%	1250	100%	54 sec
$(6960, 119, 13)$	1548	99%	1549	100%	91 sec
$(8192, 128, 13)$	1665	99%	1666	100%	105 sec

Table: Recovery of Goppa polynomial $g(x)$.

Question: What about the remaining Goppa points?

Recovering the remaining points.

Theorem Kirshanova, May (2022)

On input $H \in \mathbb{F}_2^{tm \times n}$, $(\alpha_i)_{i \in \mathcal{I}}$, $|\mathcal{I}| = tm + 1$, and $g(x)$, one recovers $(\alpha_1, \ldots, \alpha_n)$ in poly time.

Idea:

- Construct from $\mathrm{pk} = H$ a codeword $\mathbf{c}$ with $\mathrm{supp}(\mathbf{c}) \subseteq \mathcal{I} \cup \{r\}$ such that $c_r = 1$.
- Then
$$\sum_{i \in \mathcal{I}} \frac{c_i}{x - \alpha_i} \equiv -\frac{1}{x - \alpha_r} \mod g(x).$$
- Compute left-hand side, and then solve for α_r.

Recovering the remaining points.

(n, t, m)	$\ell = tm + 1$	time
$(3488, 64, 12)$	769	42 sec
$(4608, 96, 13)$	1249	130 sec
$(6960, 119, 13)$	1548	167 sec
$(8192, 128, 13)$	1665	183 sec

Table: Experimental results for point recovery.

Take Away (and compare with RSA)

- We recover the McEliece secret key with roughly $1/4$ of its bits.
- Works for random (known) positions.
- For the smallest parameter $n = 3488$ in 1 min, for the largest $n = 8192$ in < 5 mins.

LWE with Hints

- Dachman-Soled, Ducas, Gong, Rossi, "LWE with Side Information: Attacks and Concrete Security Estimation", Crypto 2020
- May, Nowakowski, "Too Many Hints — When LLL Breaks LWE", eprint 2023/777

Definition mod-q / perfect hints

LWE Public Key:	$A \in_R \mathbb{F}_q^{n \times n}$, $\mathbf{b} \in \mathbb{F}_q^n$ such that $\mathbf{b} = A\mathbf{s} + \mathbf{e}$ for small, unknown $\mathbf{s}, \mathbf{e} \in \mathbb{F}_q^n$
mod-q hints:	$\mathbf{a}_i \in_R \mathbb{F}_q^n$ and $h_i := \langle \mathbf{a}_i, \mathbf{s} \rangle \in \mathbb{F}_q$
perfect hints:	$\mathbf{a}_i \in_R \mathbb{F}_q^n$ and $h_i := \langle \mathbf{a}_i, \mathbf{s} \rangle$ over $\mathbb{Z}$ (without $\bmod q$)

Motivation:

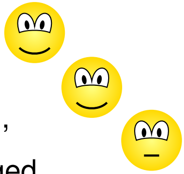
- LWE decryption computes $h := \langle \mathbf{c}, \mathbf{s} \rangle \in \mathbb{F}_q$ for ciphertexts $\mathbf{c}$.
- LWE signing computes $h := \langle H(\mathbf{m}), \mathbf{s} \rangle \in \mathbb{F}_q$ for a hashed message $H(\mathbf{m})$.

Observe: n linearly independent mod-q hints are sufficient.

Mod-q Hints

By our lattice construction, each hint

- reduces lattice dimension by one,
- reduces the shortest vector's norm,
- leaves lattice determinant unchanged.



Similarity to dlog(?)

So mod-q hints are mostly dimension reduction? No polynomial regime? Not quite.

	Kyber 512	Falcon 512	NTRU-HRSS 701	Kyber 768	Dilithium 1024
mod-q	449 (88%)	452 (88%)	622 (89%)	702 (91%)	876 (85%)
Time	20 mins	20 mins	45 mins	35 mins	10 hours

Table: Mod-q hints required for solving with LLL reduction.

Perfect Hints

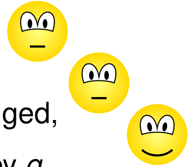
Recall: $\mathbf{a}_i \in_R \mathbb{F}_q^n$ and $h_i := \langle \mathbf{a}_i, \mathbf{s} \rangle$ over $\mathbb{Z}$

Intuition

Intuitively, perfect hints should be more powerful than mod-q hints.

By our lattice construction, each hint

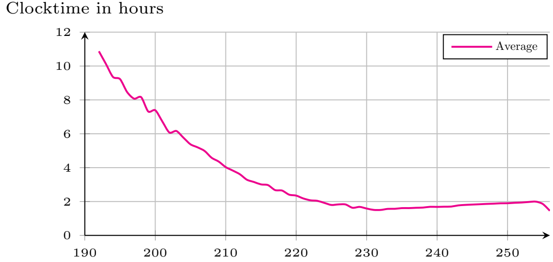
- lets lattice dimension unchanged,
- lets shortest vector's norm unchanged,
- increases the lattice determinant by q.



	Kyber 512	Falcon 512	NTRU-HRSS 701	Kyber 768	Dilithium 1024
perfect	234 (46%)	233 (46%)	332 (47%)	390 (51%)	463 (45%)
Time	3 hours	3 hours	11 hours	1 day	7 days

Table: Perfect hints required for solving with LLL reduction.

Using Stronger Lattice Reduction (BKZ)



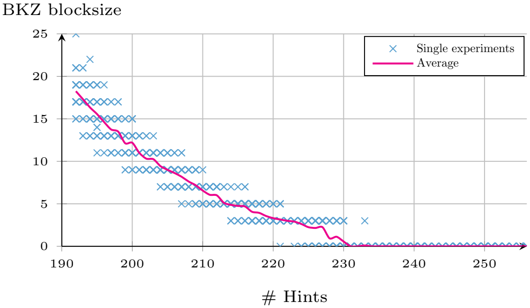


Figure: **[Kyber-512, perfect hints]** Clocktime and BKZ blocksize.

Cryptographic Key Guessing

Definition Key Guessing Problem

Let $k = k_1 \ldots k_n$ be a length-n key sampled coordinate-wise from a distribution χ: $k \leftarrow \chi^n$.
What is the number of trials to guess k?

- Consider uniform distribution $\chi = U$ with support $\{-1, 0, 1\}$. Then for all $i = 1, \ldots, n$

$$\Pr[k_i = (-1)] = \Pr[k_i = 0] = \Pr[k_i = 1] = \frac{1}{3}.$$

- $\chi = U$ has entropy

$$H(\chi) = \sum_{j \in \{-1,0,1\}} \Pr[k_i = j] \log_2\left(\frac{1}{\Pr[k_i = j]}\right) = 3 \cdot \frac{1}{3} \log_2(3) = \log_2(3) \approx 1.58.$$

- Optimal key guessing enumerates keys with at most 3^n trials, $3^n/2$ on average.
- Since $H(\chi^n) = H(\chi)n$, we express our upper bound in terms of entropy as

$$3^n = 2^{\log_2(3)n} = 2^{H(\chi)n} = 2^{H(\chi^n)}.$$

Question: Can we always guess within an entropy upper bound $2^{H(\chi)n}$?

Centered Binomial Distribution

Another example: centered binomial $\mathcal{B}(1)$

- Consider binomial distribution $\chi = \mathcal{B}(1)$ with support $\{-1, 0, 1\}$ and for all i

$$\Pr[k_i = (-1)] = \frac{1}{4}, \ \Pr[k_i = 0] = \frac{1}{2}, \ \Pr[k_i = 1] = \frac{1}{4}.$$

- Then $H(\chi) = 2 \cdot \frac{1}{4} \log(4) + \frac{1}{2} \log(2) = \frac{3}{2}$.
- So can we do within $2^{\frac{3}{2}n}$? Well, worst case still costs 3^n trials.
- Optimal algorithm: Guess keys in order of descending probability. Average case?

Lattice Standards:

- Kyber512 uses $\chi = \mathcal{B}(3)$ with support $\{-3, \dots, 3\}$.
- Kyber768 uses $\chi = \mathcal{B}(2)$ with support $\{-2, \dots, 2\}$.
- Falcon512 uses discrete Gaussian $\chi = \mathcal{D}$ with support $\{-20, \dots, 20\}$.
- Falcon1024 uses discrete Gaussian $\chi = \mathcal{D}$ with support $\{-13, \dots, 13\}$.

Why do we actually think of entropy?

Intuition for entropy bound from Information Theory

- Any key $k \leftarrow \chi^n$ can be compressed lossless to $(H(\chi) + \epsilon)n$ bits, ϵ constant.
- Algorithm: Enumerate compressed keys instead of keys themselves.
- Leads to an algorithm with $2^{H(\chi)n} \cdot 2^{\epsilon n}$ trials.
- MATZOV(22) used $2^{H(\chi)n}$ for analyzing lattice-based schemes. Large underestimate?

Experimental Evidence

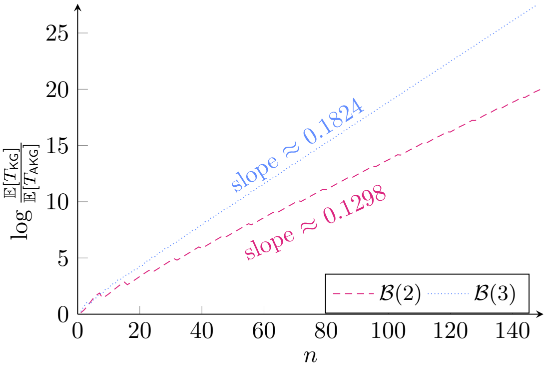


Figure: $\epsilon n = \log(\mathbb{E}[T]/2^{H(\chi)n})$

Similar results:

- Albrecht, Shen, "Quantum Augmented Dual Attack", 2022
- Ducas, Pulles, "Does the Dual-Sieve Attack on LWE even Work?", 2023

Our Result

Complexity Measure

So far we sticked to success probability $p = 1$, and tried to bound the number of trials t. Why not optimize t/p?

Aborted Key Guessing: Abort when the probability for next key guess hits a threshold.

Theorem Glaser, May, Nowakowski, eprint 2023/797

For *any distribution* χ, Aborted Key Guessing

1. uses at most $t \leq 2^{H(\chi)n}$ trials,
2. has success probability $p \to \frac{1}{2}$,
3. allows for (optimal) quantum-type Grover version with $t \leq 2^{H(\chi)n/2}$ trials.

Convergence Experimentally

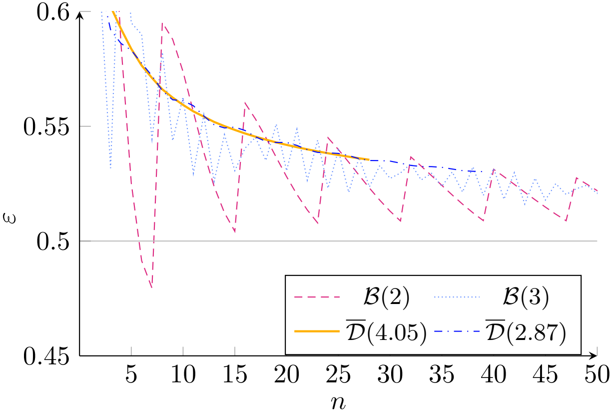


Figure: Convergence of success probability $p \to \frac{1}{2}$

Lessons Learned

- Some schemes allow for efficient Partial Key Exposure: RSA, McEliece.
- McEliece attack works with less information, in random positions, and faster.
- Put positively, one can compress a McEliece secret key to $1/4$ of its size.

- Some schemes seem more resistant to Partial Key Exposure: Dlog, lattices.
- Lattices still allow for Partial Key Exposures beyond pure dimension reduction.
- Key redundancy seems to play major role for Partial Key Exposure attacks.

- Key Guessing can be done within $2^{H(\chi)n}$ trials for any χ with probability $p \to \frac{1}{2}$.

Question:

- Which information do we obtain from real side-channels?