Building and Breaking Lattice-Based Post-Quantum Cryptosystem Hardware

Aydin Aysu, Assistant Professor

Researchers: Dr. Seetal Potluri, Anuj Dubey, Furkan Aydin, Emre Karabulut, Priyank Kashyap, Gregor Haas, Faiz Alam, Ferhat Yaman

aaysu@ncsu.edu

NC STATE

Research Sponsors

SRC® Semiconductor Research Corporation

NSF

CAEML CENTER FOR ADVANCED ELECTRONICS THROUGH MACHINE LEARNING

Office of Naval Research Science & Technology

Our Research

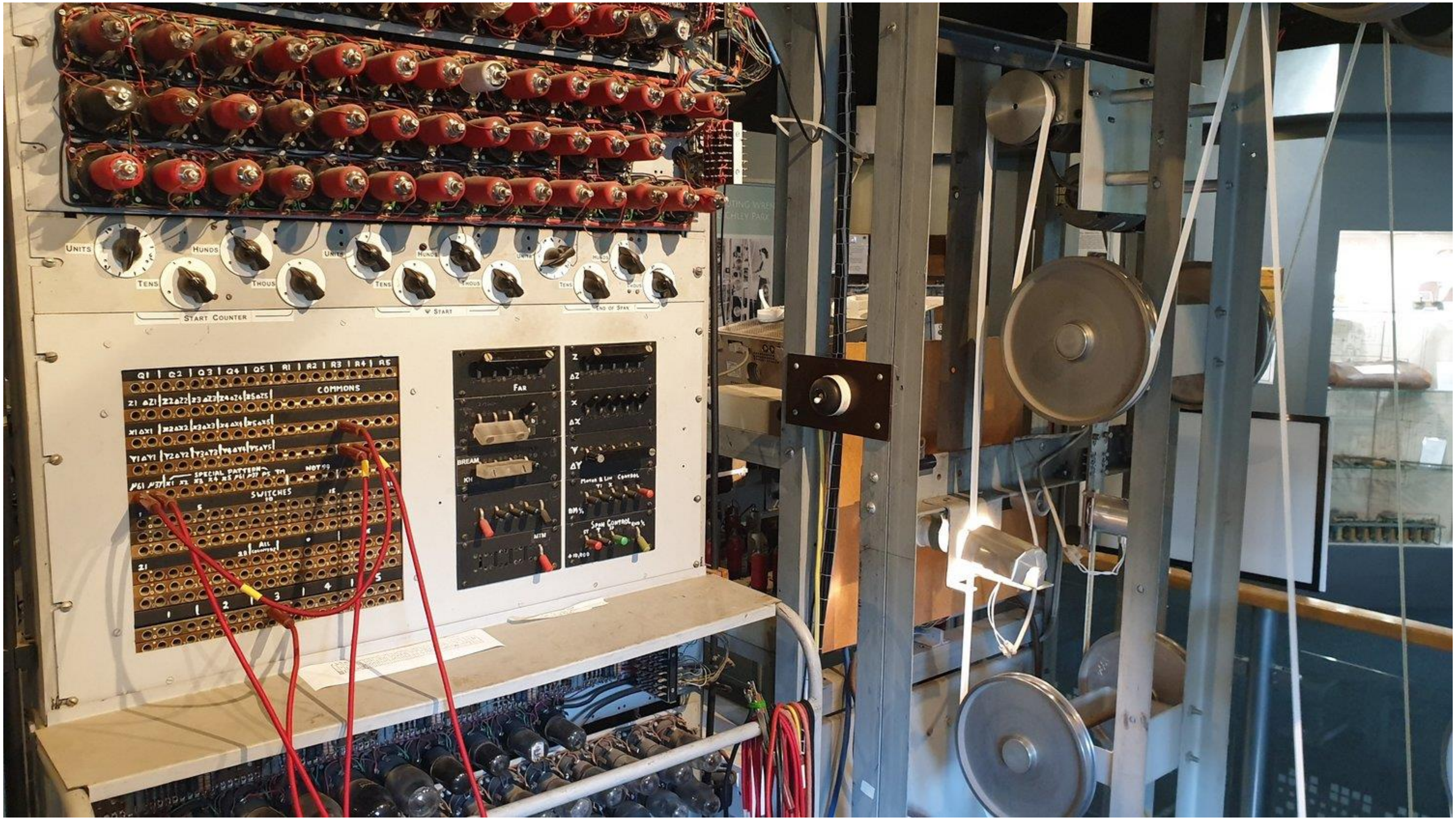
Cybersecurity with a ***hardware*** focus

- Hardware acceleration for next-generation cryptography
  [DATE’20][TC’20][FPL’20][ICCAD’20][ESL’19][TC’15][TECS’15][ESL’14] **[HOST’13]**
- Hardware building blocks to combat supply-chain attacks
  [ICCAD’21] [HOST’18][HOST’17][ICISC’16][DATE’16][CHES’15][WESS’13]
- Mitigating hardware theft of untrusted foundries
  [TCAD’22][ISQED’20][ISCAS’20][TCAD’22]
- Implementation security: side-channel and fault attacks
  [DAC’22]**[HOST’22][TCHES’22]**[DAC’21][HOST’20][ICCAD’20]**[HOST’18]**[DATE’14]
- Training a cyber-aware STEM workforce
  [GLS-VLSI’22][GLS-VLSI’19]

Why Quantum Computing?

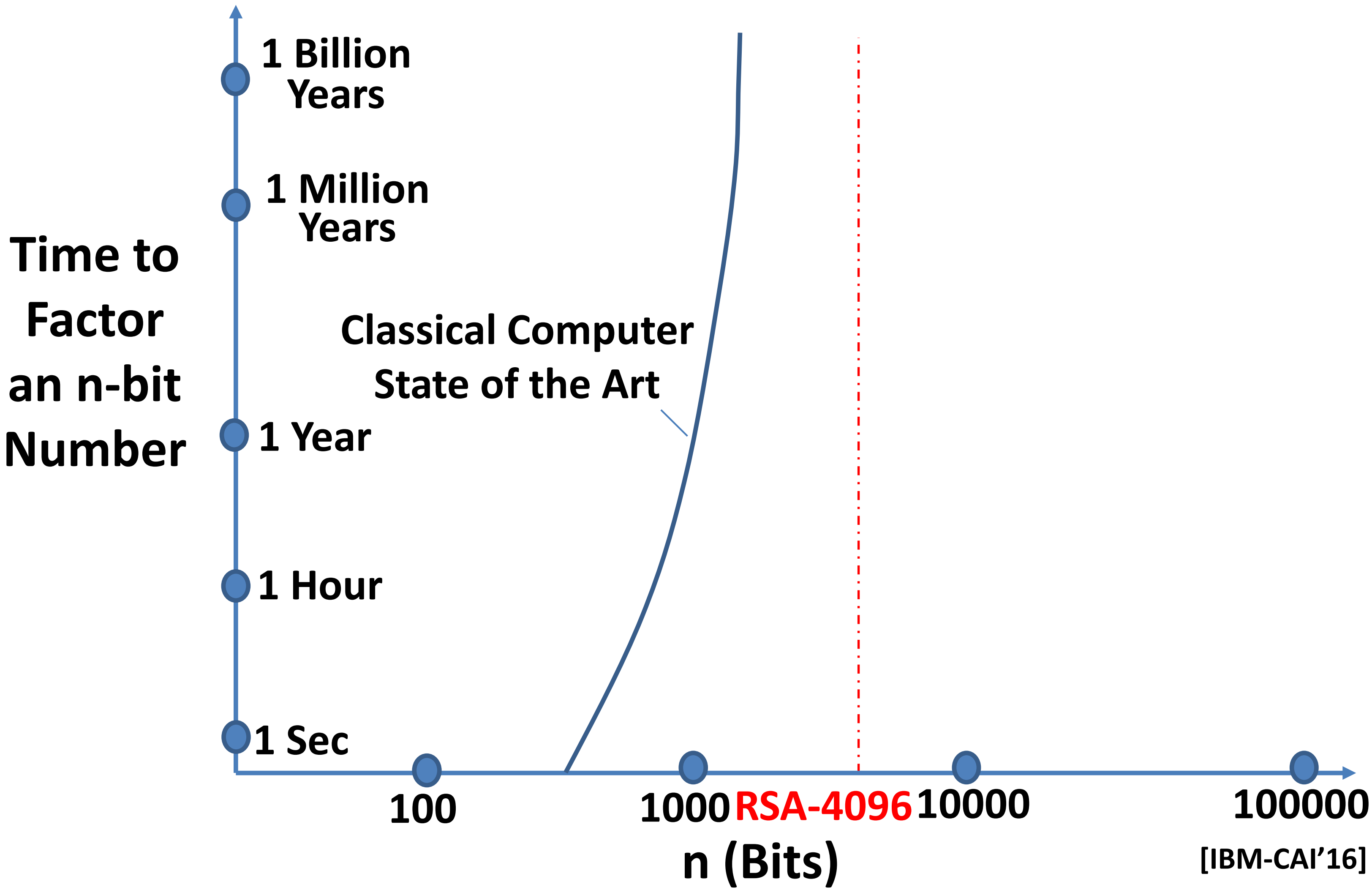
- Better predicting tomorrow's weather?
- Efficient simulation of chemical reactions?
- Finding new electronic materials?
- Optimize traffic, logic simulations, or ticket prices?
- …

Know This Machine?



Quantum Computers Endanger Cryptography

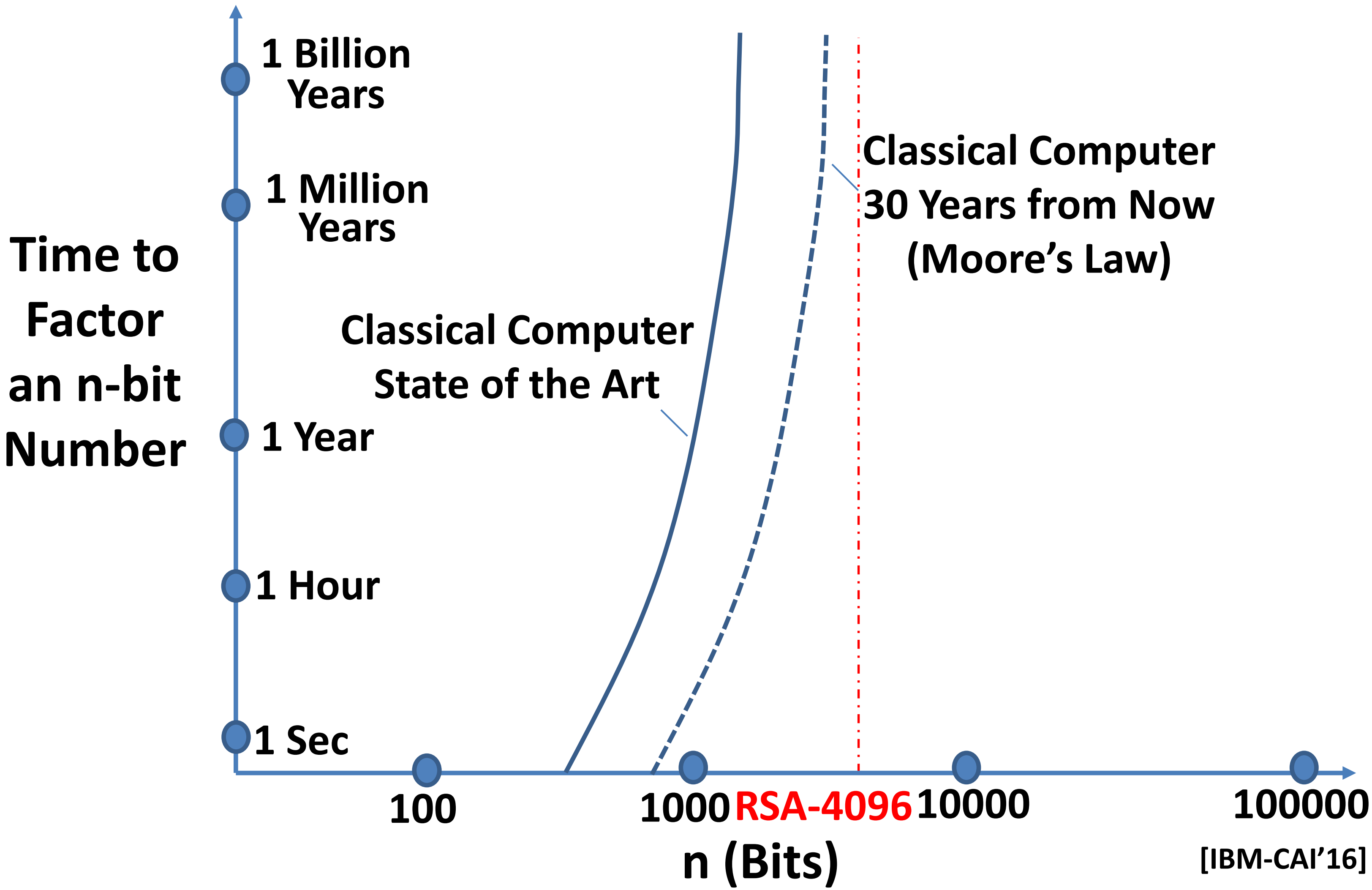
Encryption for the web rely on hard mathematical problems



[IBM-CAI'16]

Quantum Computers Endanger Cryptography

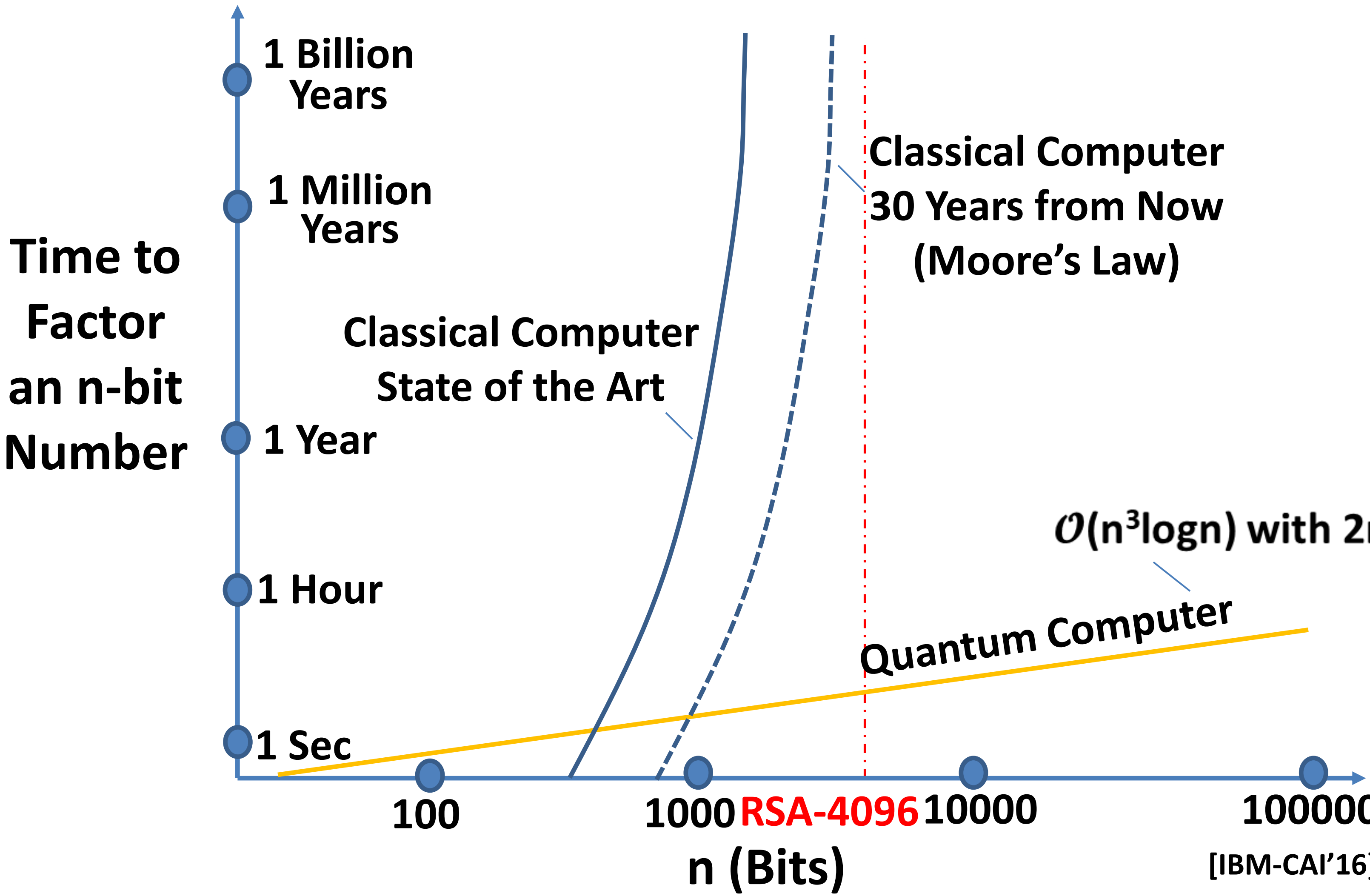
Encryption for the web rely on hard mathematical problems



[IBM-CAI'16]

Quantum Computers Endanger Cryptography

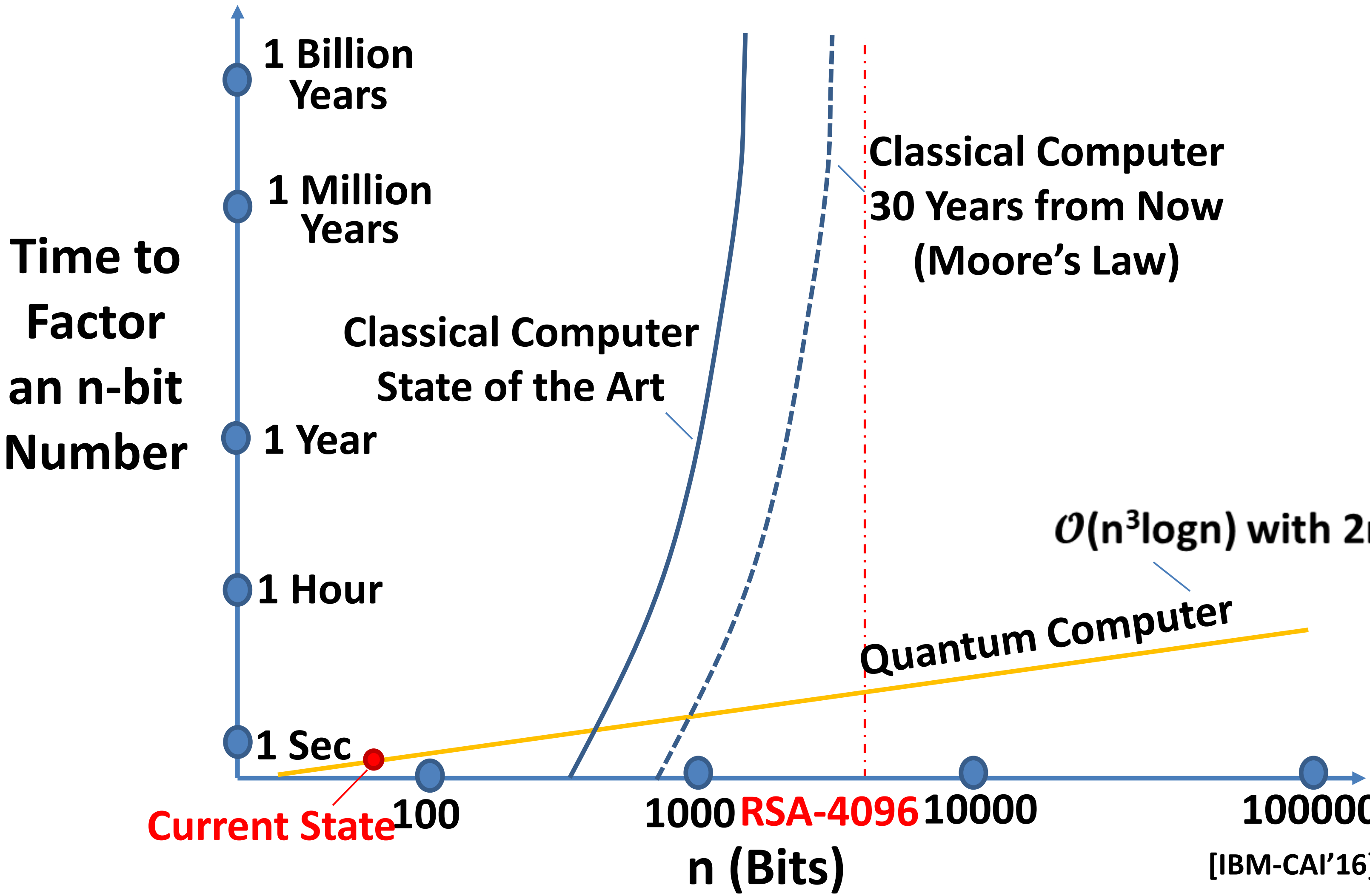
Encryption for the web rely on hard mathematical problems



[IBM-CAI'16]

Quantum Computers Endanger Cryptography

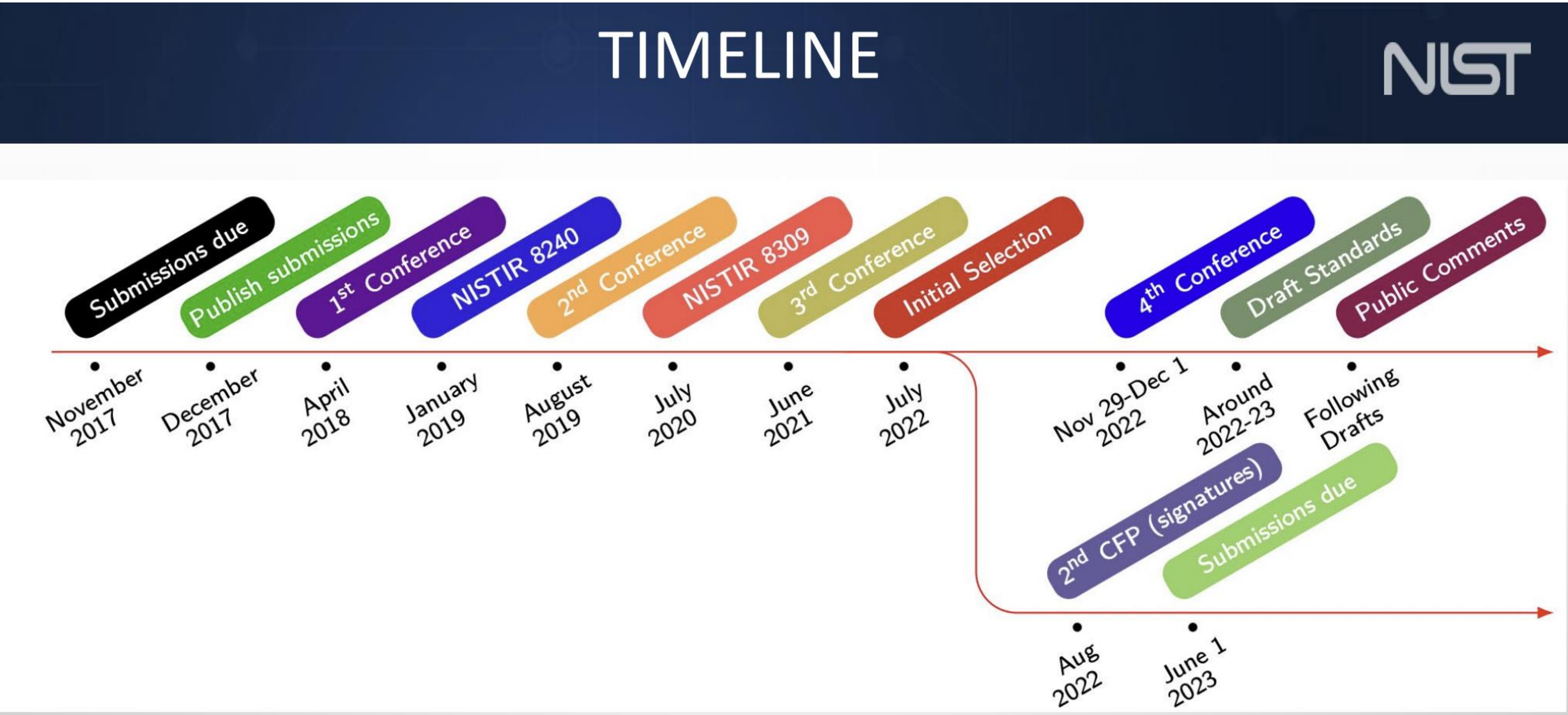
Encryption for the web rely on hard mathematical problems



[IBM-CAI'16]

Emergence of Post-Quantum Cryptography

- NIST's PQ standardization effort (2017–2024)
- Some industry/government adoptions already occurred



Emergence of Post-Quantum Cryptography

- Key Encapsulation Mechanisms
  - **CRYSTALS-KYBER**
- Digital Signatures
  - **CRYSTALS-DILITHIUM**
  - **FALCON**
  - SPHINCS+
- Alternates:
  - **FrodoKEM, NTRU, NTRU Prime, SABER, …**

Moving to Quantum-Secure Cryptography

NIST | NATIONAL CYBERSECURITY CENTER OF EXCELLENCE

SECURITY GUIDANCE

Migration to Post-Quantum Cryptography

The advent of quantum computing technology will compromise many of the current cryptographic algorithms, especially public-key cryptography, which is widely used to protect digital information. Most algorithms on which we depend are used worldwide in components of many different communications, processing, and storage systems. Once access to practical quantum computers becomes available, all public-key algorithms and associated protocols will be vulnerable to criminals, competitors, and other adversaries. It is critical to begin planning for the replacement of hardware, software, and services that use public-key algorithms now so that information is protected from future attacks.

Collaborating Vendors

SEARCH

- Amazon Web Services, Inc. (AWS)
- Cisco Systems, Inc.
- Crypto4A Technologies, Inc.
- CryptoNext Security
- Dell Technologies
- DigiCert
- Entrust
- IBM
- InfoSec Global
- ISARA Corporation
- JPMorgan Chase Bank, N.A.
- Microsoft
- Samsung SDS Co., Ltd.
- SandboxAQ
- Thales DIS CPL USA, Inc.
- Thales Trusted Cyber Technologies
- VMware, Inc.
- wolfSSL

Source: https://www.nccoe.nist.gov/

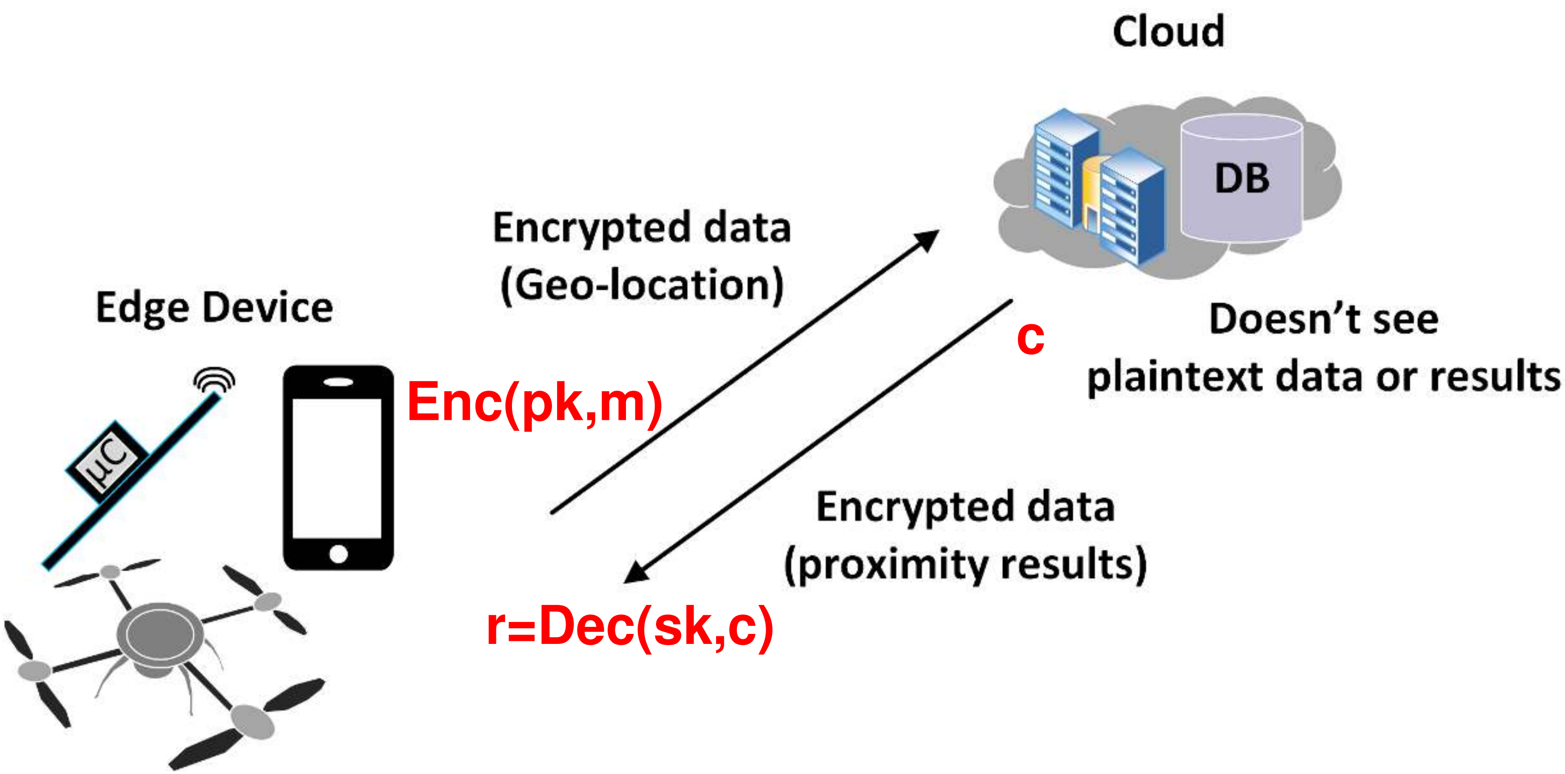
Category of Post-Quantum Cryptosystems

Category	Security Assumption	Features
Code-based cryptography	**Decoding general linear codes**	Large keys, complex operations
Hash-based cryptography	**One-way hash functions**	Large keys, limited applications
Lattice-based cryptography	**Lattice problems**	Small keys, efficient arithmetic,

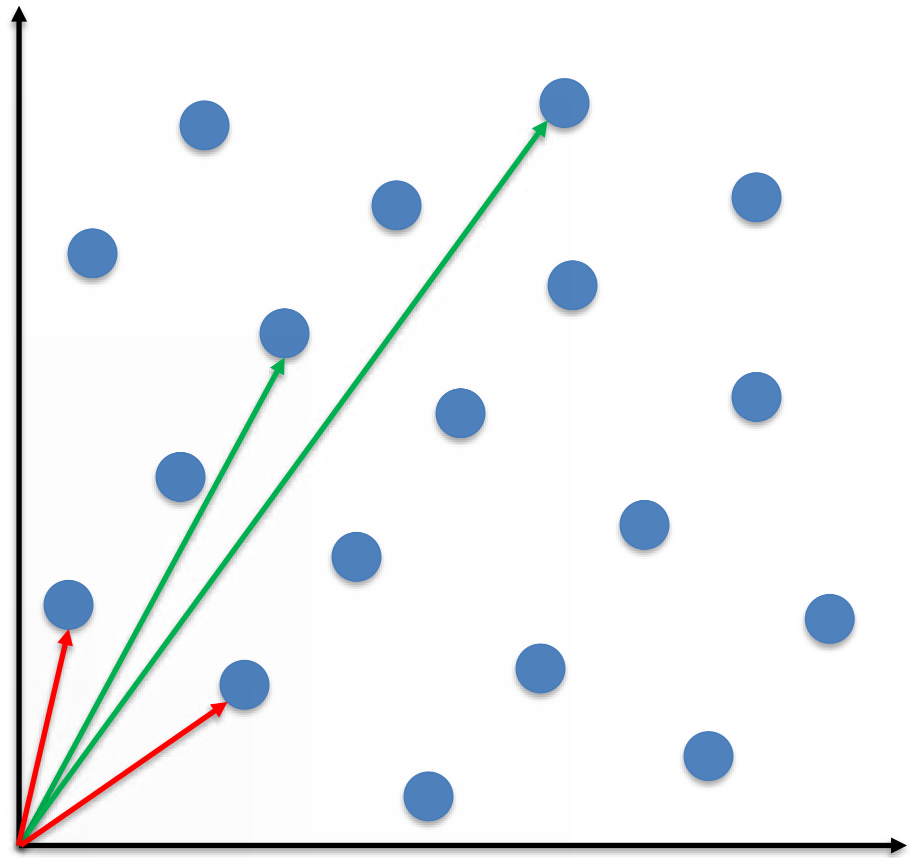
3 out of the 4 upcoming NIST standards use lattice cryptography

Lattices Have Other Uses…

Homomorphic encryption allows computing on encrypted data <u>without</u> knowing the secret key or underlying plaintext



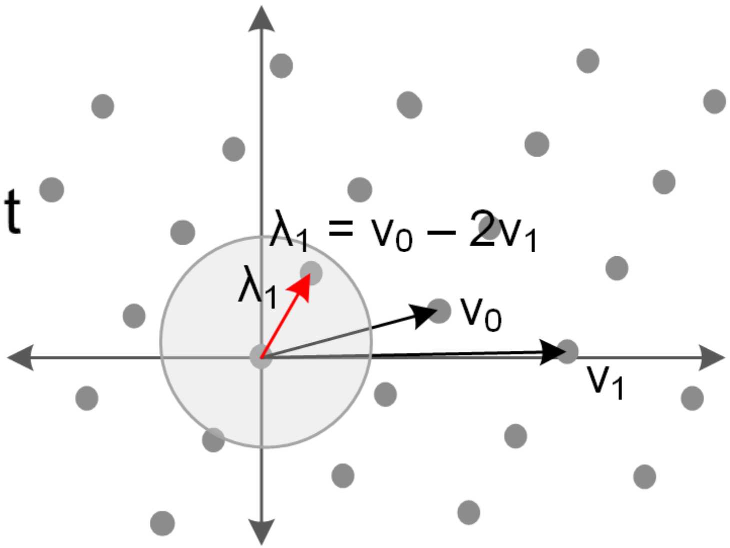
Security of Lattice-Based Cryptography



Given a bad basis, can you find a good one?

Lattice-based Cryptography

- A Lattice is a set of points
  $L=\{a_1v_1+\ldots+a_nv_n \mid a_i \text{ integers}\}$
  with $v_1,\ldots,v_n$ in R^n linearly independent

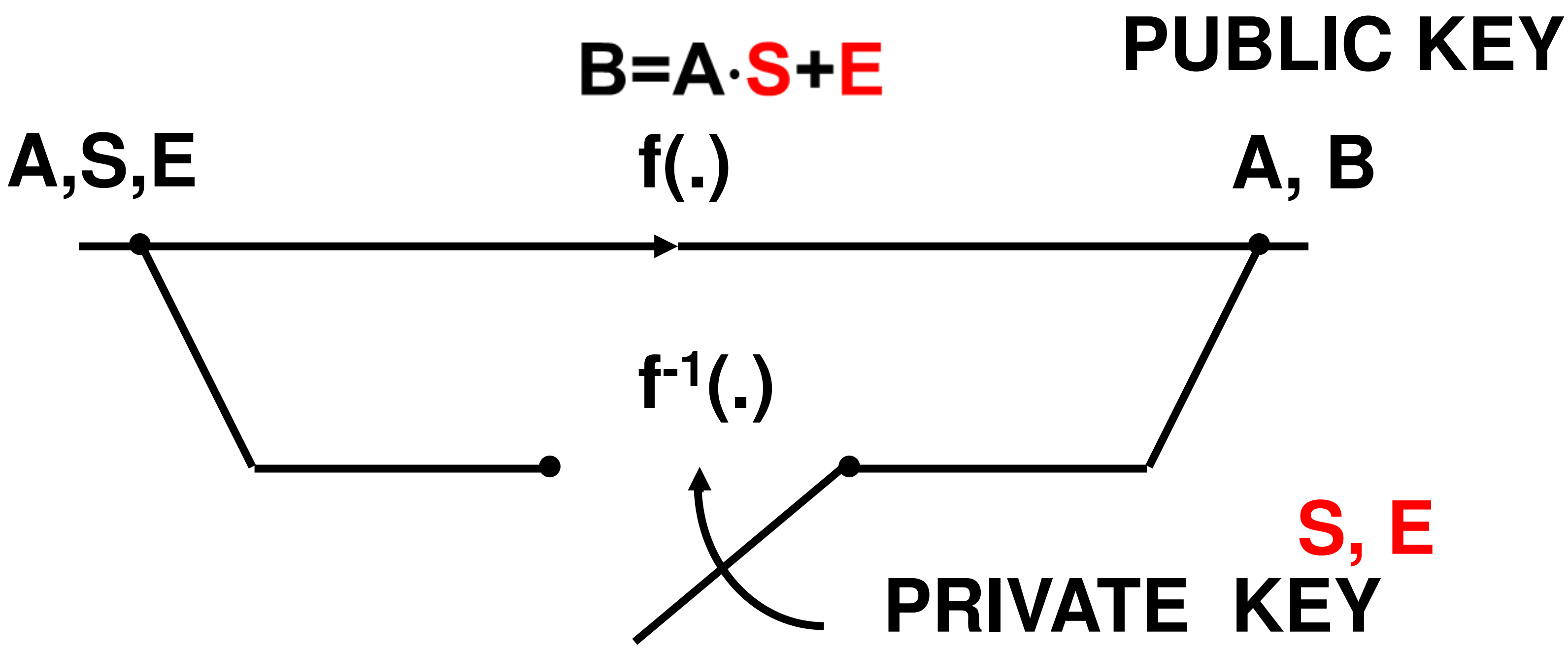


- Approximate Shortest Vector Problem (SVP): Given basis v_0,v_1 find a short vector λ_1
- NP-Hard [Ajtai'96]
- Lattice basis reduction attack complexities
  - Classical: $2^{2n+o(n)}$ [MV'10]
  - Quantum: $2^{1.799n+o(n)}$ [LMP'13]

Trap-door one-way function

Learning With Errors:

B=A·S+E, PUBLIC KEY=B and A, SECRET KEY=S



Works with matrices and polynomials

Fundamental Computations in Lattice-Based Cryptography

Matrix Multiplication

$$\underset{A}{\begin{bmatrix} 2 & 13 & 7 & 3 \\ 4 & 7 & 9 & 1 \\ 6 & 14 & 5 & 11 \end{bmatrix}} \bullet \underset{S}{\begin{bmatrix} 8 \\ 3 \\ 12 \\ 5 \end{bmatrix}} + \underset{E}{\begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}} = \underset{B}{\begin{bmatrix} 13 \\ 12 \\ 3 \end{bmatrix}}$$

Polynomial Multiplication

$$\underset{a}{\begin{bmatrix} 2 \\ 13 \\ 7 \\ 3 \end{bmatrix}} * \underset{s}{\begin{bmatrix} 8 \\ 3 \\ 12 \\ 5 \end{bmatrix}} + \underset{e}{\begin{bmatrix} 1 \\ -1 \\ 2 \\ -1 \end{bmatrix}} = \underset{b}{\begin{bmatrix} 8 \\ 1 \\ 16 \\ 6 \end{bmatrix}}$$

Elements are defined over Galois Field (modular arithmetic with primes)
Random sampling may require “discrete Gaussian” distributions

FALCON Specification – What to Implement?

Algorithm 5 NTRUGen(ϕ, q)

Require: A monic polynomial $\phi \in \mathbb{Z}[x]$ of degree n, a modulus q

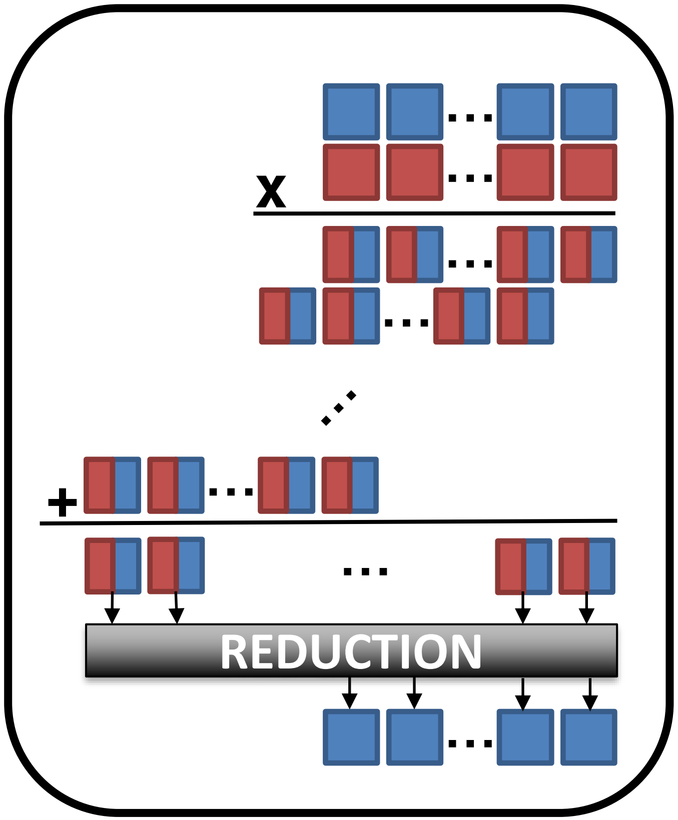
Ensure: Polynomials f, g, F, G

- $\sigma_{\{f,g\}} \leftarrow 1.17\sqrt{q/2n}$ ▷ $\sigma_{\{f,g\}}$ is chosen so that $\mathbb{E}[\|(f,g)\|] = 1.17\sqrt{q}$
- **for** i from 0 to $n-1$ **do**
  - $f_i \leftarrow D_{\mathbb{Z},\sigma_{\{f,g\}},0}$ ▷ See also (3.29)
  - $g_i \leftarrow D_{\mathbb{Z},\sigma_{\{f,g\}},0}$
- $f \leftarrow \sum_i f_i x^i$ ▷ $f \in \mathbb{Z}[x]/(\phi)$
- $g \leftarrow \sum_i g_i x^i$ ▷ $g \in \mathbb{Z}[x]/(\phi)$
- **if** NTT(f) contains 0 as a coefficient **then** ▷ Check that f is invertible mod q
  - restart
- $\gamma \leftarrow \max\left\{\|(g,-f)\|, \left\|\left(\frac{qf^\star}{ff^\star+gg^\star}, \frac{qg^\star}{ff^\star+gg^\star}\right)\right\|\right\}$ ▷ Using (3.9) with (3.8) or (3.10)
- **if** $\gamma > 1.17\sqrt{q}$ **then** ▷ Check that $\gamma = \|\mathbf{B}\|_{GS}$ is short
  - restart
- $F, G \leftarrow$ NTRUSolve$_{n,q}(f, g)$ ▷ Computing F, G such that $fG - gF = q \bmod \phi$
- **if** $(F, G) = \perp$ **then**
  - restart
- **return** f, g, F, G

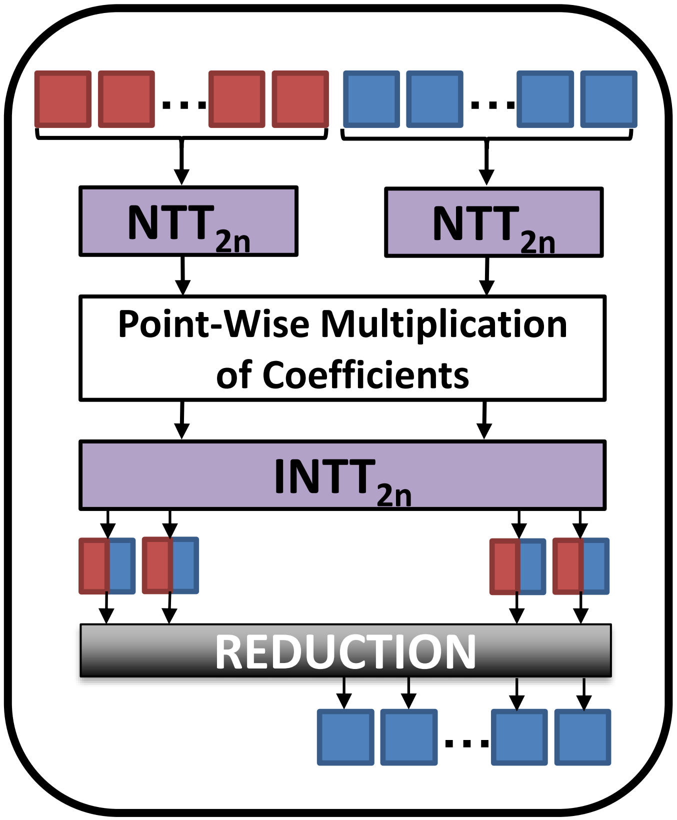
New IPs Needed for Lattice-Based Cryptography

- Building blocks for discrete Gaussian sampling
- Building blocks for Number Theoretic Transform
- Full system design working with new building blocks
- System-level trade-offs
- Optimizations for edge computers to cloud
- New custom instructions for ISA
- Implementation security!
- Hybrid designs

Number Theoretic Transform



Schoolbook Method



NTT-based Method

Reduces multiplication complexity from $\boldsymbol{O}(\mathbf{n}^2)$ to $\boldsymbol{O}(\mathbf{n.logn})$

Number Theoretic Transform

Iterative NTT Algorithm

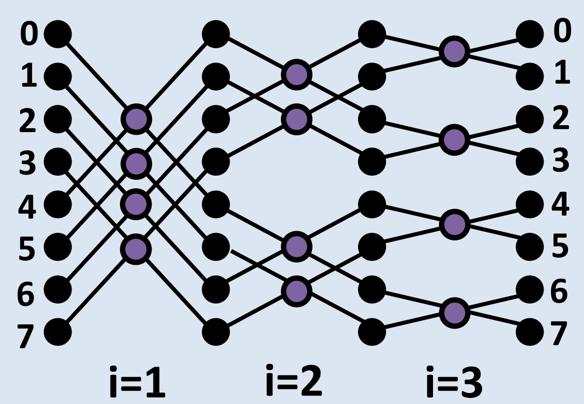
Algorithm 2 Iterative NTT Algorithm [14]

Input: $A(x) \in \mathbb{Z}_q[x]/(x^n + 1)$
Input: primitive n-th root of unity $\omega \in \mathbb{Z}_q$, $n = 2^l$
Output: $\overline{A}(x) = \mathbf{NTT}(A) \in \mathbb{Z}_q[x]/(x^n + 1)$

for i from 1 by 1 to l **do**
  $m = 2^{l-i}$
  for j from 0 by 1 to $2^{i-1} - 1$ **do**
    for k from 0 by 1 to $m - 1$ **do**
      $U \leftarrow A[2 \cdot j \cdot m + k]$
      $V \leftarrow A[2 \cdot j \cdot m + k + m]$
      $A[2 \cdot j \cdot m + k] \leftarrow U + V$
      $A[2 \cdot j \cdot m + k + m] \leftarrow \omega^{(2^{i-1}.k)} \cdot (U - V)$
    end for
  end for
end for
return A

Read
Butterfly
Write

8-point NTT



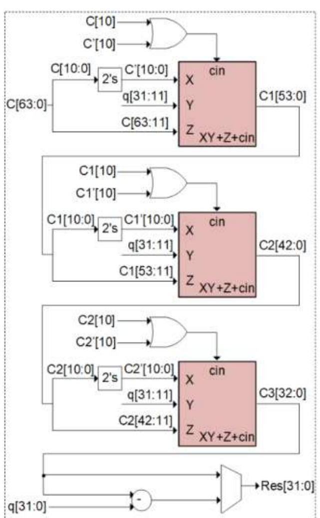
- ❑ N-point NTT operation has $\log_2 n$ stages
- ❑ At each stage, n/2 butterfly operation is performed
- ❑ Single NTT operation can be parallelized using multiple butterfly units

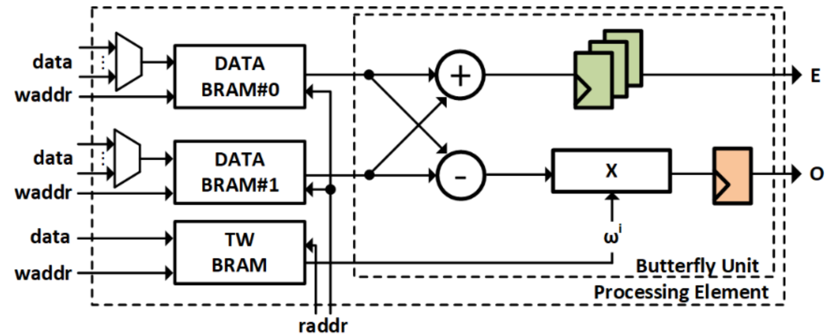
Number Theoretic Transform

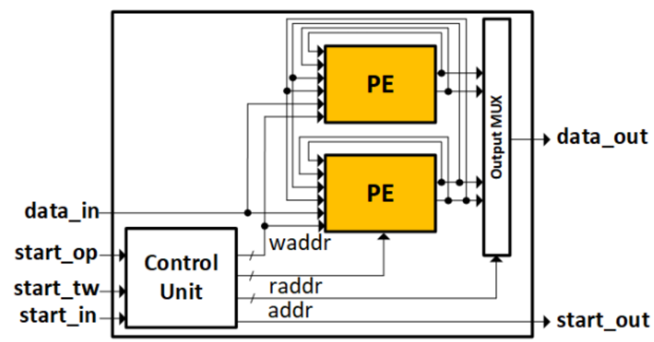
Algorithm 3 Word-Level Montgomery Reduction Algorithm for NTT-friendly primes

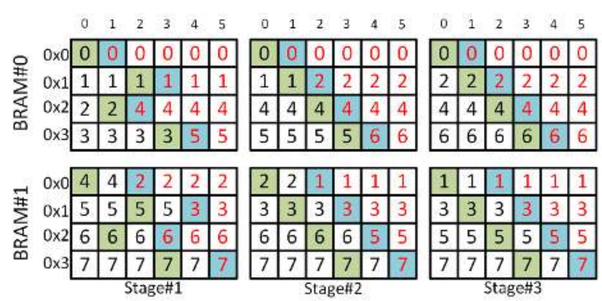
Input: $C = A \cdot B$ (a $2K$-bit positive integer)
Input: q (a K-bit modulus), $q = q_H \cdot 2^w + 1$
Input: $w = log_2(2n)$ (word size)
Output: $Res = C \cdot R^{-1} \pmod q$ where $R = 2^{w \times L} \pmod q$

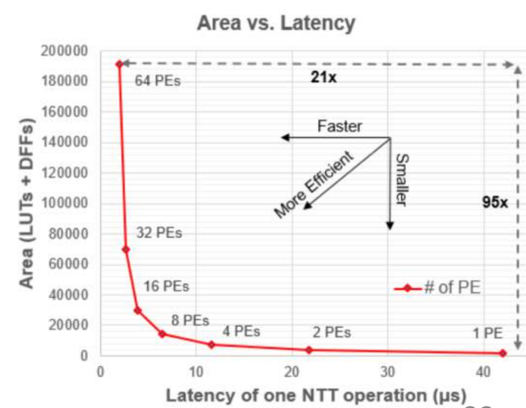
1: $L = \lceil \frac{K}{w} \rceil$
2: $T1 = C$
3: **for** i from 0 to L **do**
4: $\quad T1_H = T1 >> w$
5: $\quad T1_L = T1 \pmod{2^w}$
6: $\quad T2 = two's\ complement\ of\ T1_L$
7: $\quad cin = T2[w-1] \vee T1_L[w-1]$
8: $\quad T1 = T1_H + (q_H \cdot T2[w-1:0]) + cin$
9: **end for**
10: $T4 = T1 - q$
11: **if** $(T4 < 0)$ **then** $Res = T1$ **else** $Res = T4$





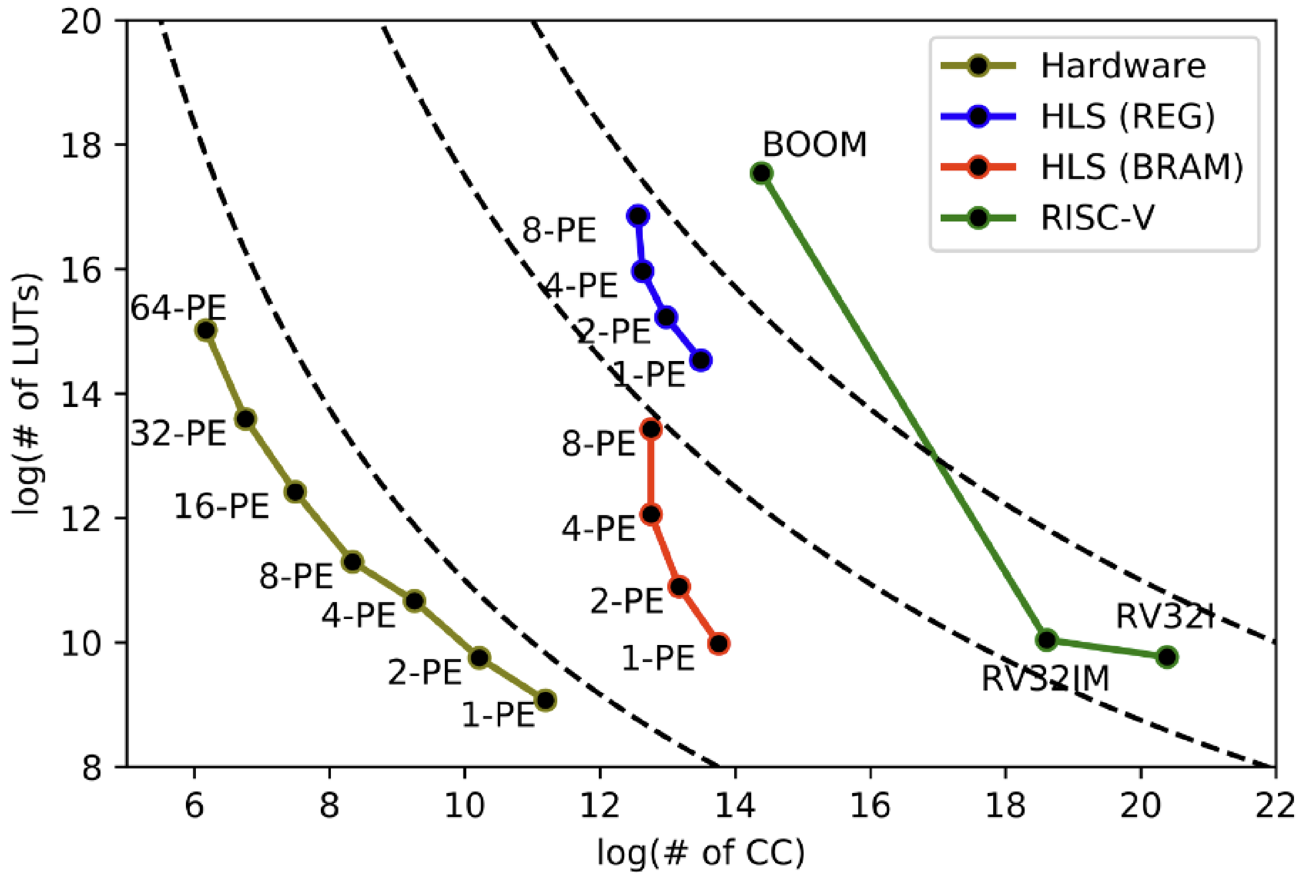






Aydin Aysu et al. "An extensive study of flexible design methods for the number theoretic transform." *IEEE Transactions on Computers* 71, no. 11 (2020): 2829-2843.

Number Theoretic Transform Results



Aydin Aysu et al. "An extensive study of flexible design methods for the number theoretic transform." *IEEE Transactions on Computers* 71, no. 11 (2020): 2829-2843.

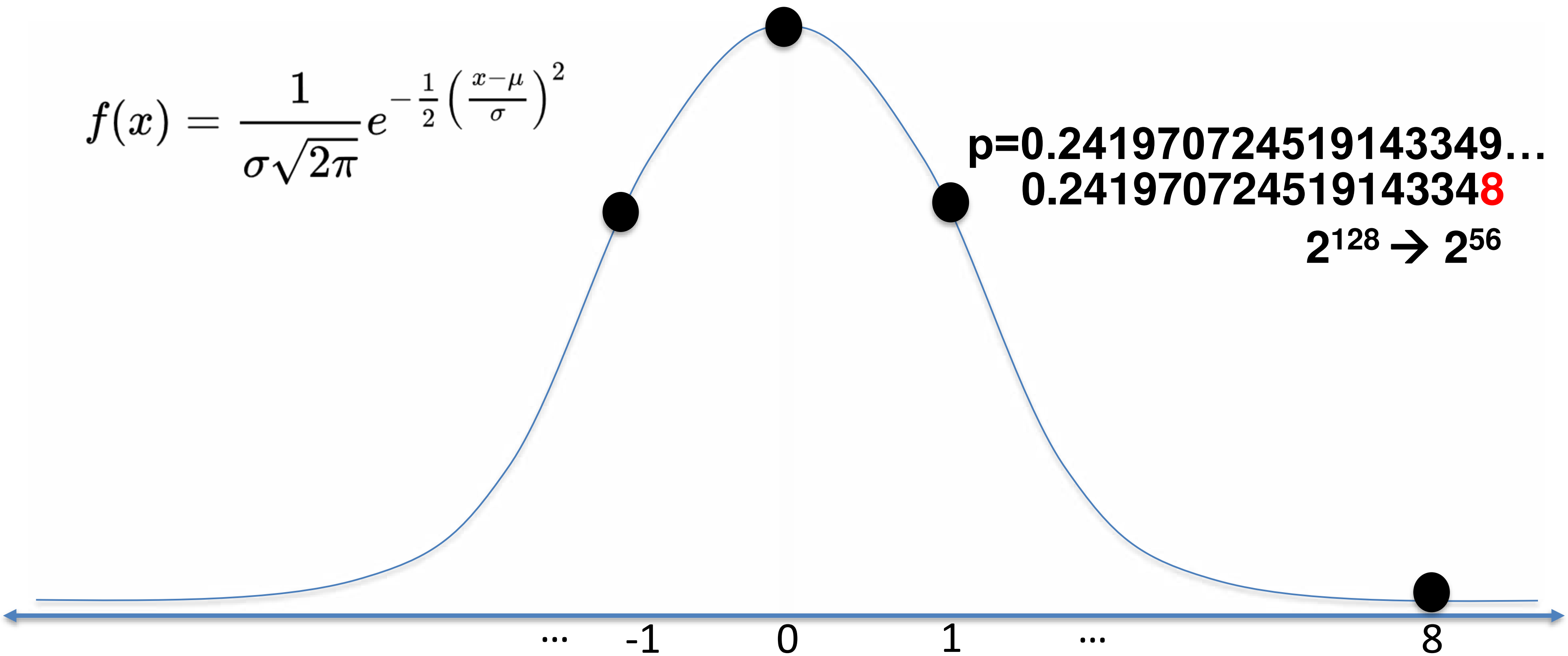
Number Theoretic Transform Results

Met.	Work	Platform	n	K	LUT / REG / DSP / BRAM	Clock (MHz)	Latency CC	Latency μs
Hardware	[20][a]	Spartan-6	256 512 1024	17	250 / – / 3 / 2 240 / – / 3 / 2 250 / – / 3 / 2	–	– – –	25 50 100
	[21][a,b]	Virtex-6	256	13	4549 / 3624 / 1 / 12	262	–	8
	[22][b]	Zynq US	4096	30	64K / – / 200 / 400	225	–	73
	[23][b]	Virtex-7	32768	32	219K / – / 768 / 193	250	7709	51
	[24][b]	Spartan-6 Virtex-7	1024	32	1208 / – / 14 / 14 34K / 16K / 476 / 228	212 200	– 80	12 0.4
	[18][b]	Virtex-7	1024	32	67K / – / 599 / 129 77K / – / 952 / 325.5	200	140 80	0.7 0.4
	[25][c]	Virtex-6	256 512	13 14	1349 / 860 / 1 / 2 1536 / 953 / 1 / 3	313 278	1691 3443	5.4 12.3
	[11][c]	40nm CMOS	256 512 1024	13 14 14	106K / – / – / –	72	1289 2826 6155	17 32 81
	[26][c]	40nm CMOS	256 512	13 14	– / – / – / – – / – / – / –	300	160 492	0.5 1.6
	[27][c]	UMC 65nm	256 512 1024	13 14 14	14K / – / – / –	25	2056 4616 10248	82 184 409
	[28][a,b]	Artix-7	1024	14	4823 / 2901 / 8 / –	153	1280	–
	[29][b]	Virtex-7	16384 32768	32 32	2.81K / 1.25K / 39 / 80 2.86K / 1.27K / 39 / 160	168 166	28672 61440	– –
	[30][b]	Virtex-6	65536	30	72K / 63K / 250 / 84	100	47795	–
	TW-1 PE	Virtex-7	1024 4096	14 60	575 / – / 3 / 11 2720 / – / 31 / 180	125	5160 24708	41.2 197.6
	TW-8 PE	Virtex-7	1024 4096	14 60	2584 / – / 24 / 16 23215 / – / 248 / 176	125	680 3276	5.4 26.2
	TW-32 PE	Virtex-7	1024 4096	14 60	17188 / – / 96 / 48 99384 / – / 992 / 176	125	200 972	1.6 7.7

Aydin Aysu et al. "An extensive study of flexible design methods for the number theoretic transform." *IEEE Transactions on Computers* 71, no. 11 (2020): 2829-2843.

High Precision Discrete Gaussian Sampling

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$



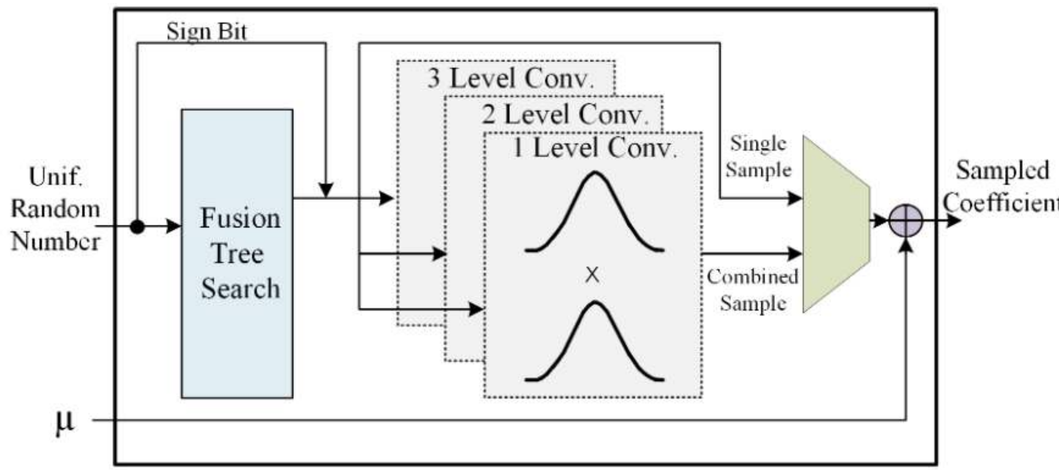
Sampling precision impacts cryptographic security level

High Precision Discrete Gaussian Sampling

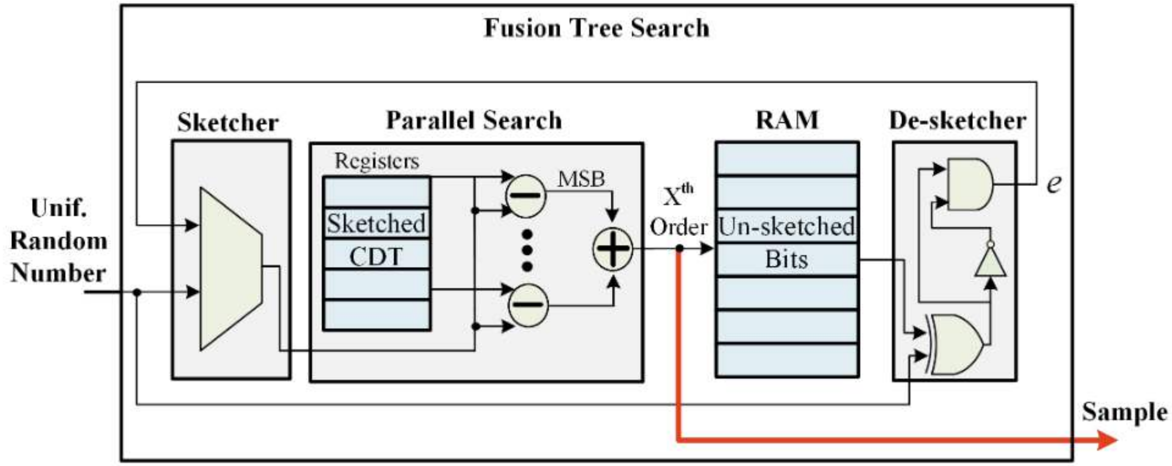
Sampler	Speed	FP exp()	Table Size	Table Lookup	Entropy	Features
Rejection	slow	10	0	0	$45{+}10log_2\sigma$	Suitable for constrained devices
Ziggurat	flexible	flexible	flexible	flexible	flexible	Suitable for encryption requires high-precision FP arithmetic; not suitable for HW implementation
CDT	fast	0	$\sigma\tau\lambda$	$log_2(\tau\sigma)$	$2.1{+}log_2\sigma$	Suitable for digital signature easy to implement
Knuth-Yao	fastest	0	$1/2\sigma\tau\lambda$	$log_2(\sqrt{2\pi e}\sigma)$	$2.1{+}log_2\sigma$	Not suitable for digital signature
Bernoulli	fast	0	$\lambda log_2(2.4\tau\sigma^2)$	$\approx log_2\sigma$	$\approx 6 + 3log_2\sigma$	Suitable for all schemes
Binomial	fast	0	0	0	$4\sigma^2$	Not suitable for digital signature

Many algorithmic options for implementing Gaussian sampling

High Precision Discrete Gaussian Sampling



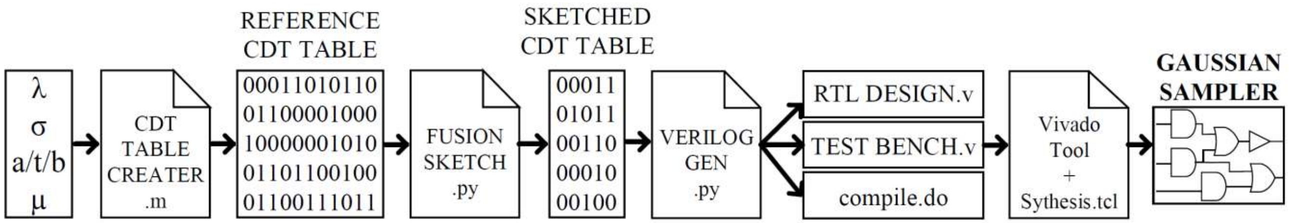
(a) The Proposed Gaussian Sampler Hardware's Top Level Block Diagram



(b) Fusion Tree Search's Block Diagram

Aydin Aysu et al. "Efficient, flexible, and constant-time gaussian sampling hardware for lattice cryptography." *IEEE Transactions on Computers* 71, no. 8 (2021): 1810-1823.

Results and Comparison

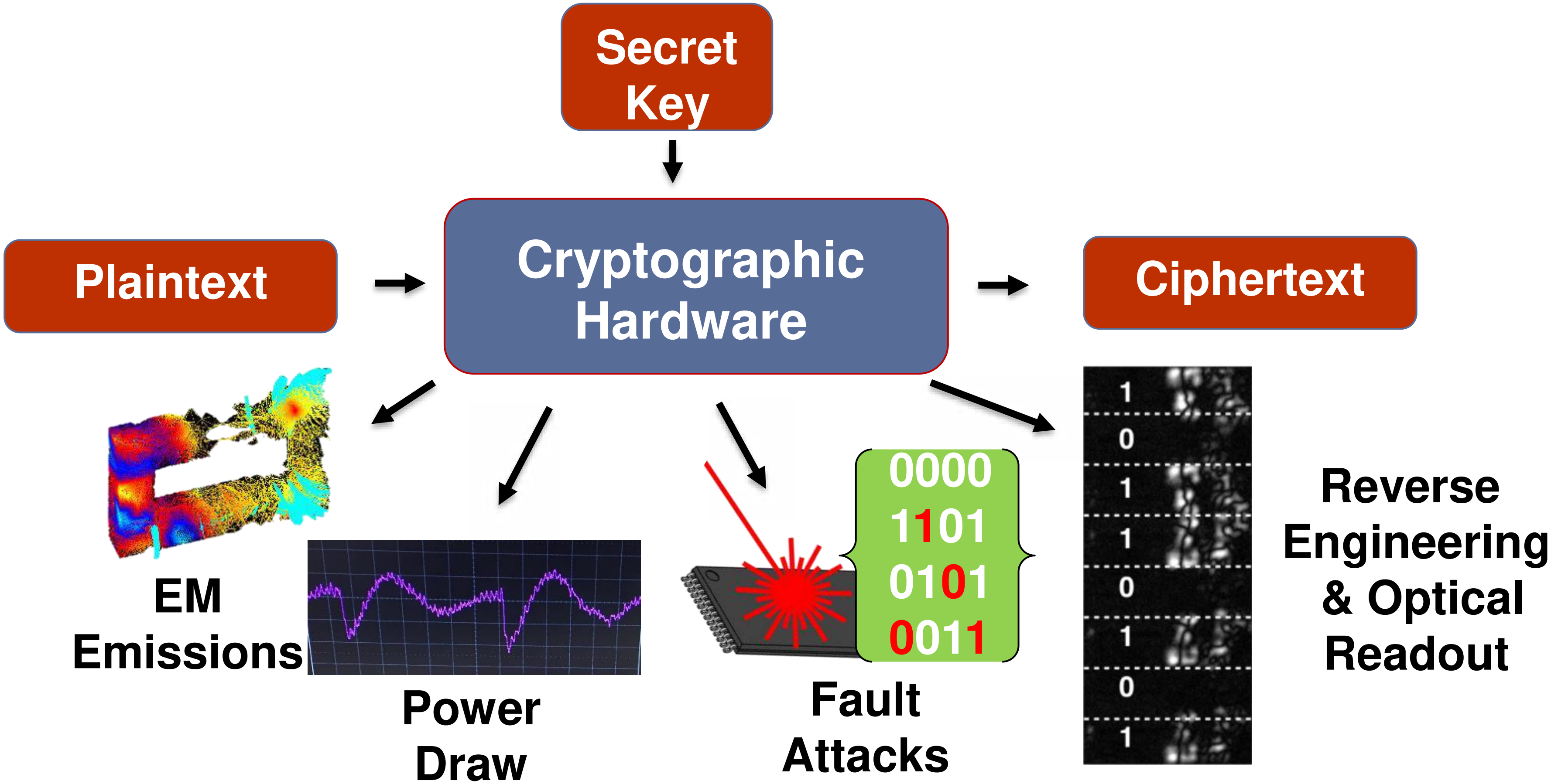


Work	Supported Algorithms	$\sigma/\lambda/Depth$	Platform	Slice/LUTs/ FFs/BRAM	F_{Max} (MHz)	Cyc Cnt	Area-Delay
HW [15]	qTESLA p-I	8.5/64/77	Artix-7	-/907/812/3	115	111	235.88×
This Work[a]		8.5/64/80	Virtex-7	169/554/306/0	232	3	-
HW [15]	qTESLA p-III	8.5/125/110	Artix-7	-/820/837/3	119	49	35.26×
This Work[a]		8.5/128/112	Virtex-7	324/1049/566/0	162	3	-
HW [11]	LP	3.33/64/31	Virtex-6	43/112/19/0	297	5	0.16×
HW [10]		3.33/90/37	Virtex-5	17/43/33/1	259	3	0.36×
HW [8]		3.33/80/35	Virtex-6	231/863/6/0	61	1	1.06×
This Work	LP	3.33/64/33	Virtex-7	360/1278/306/0	218	2	-
		3.33/90/37		442/1418/306/0	198	2	
		3.33/80/35		425/1341/8/0	205	2	
		3.33/100/39		539/1960/446/0	173	2	
HW[a] [11]	BLISS	215/64/184	Spartan-6	179/577/64/0	130	8	1.67×
HW[a] [13]		215/128/184	Spartan-6	299/928/1121/0	129	8	2.85×
This Work[a]		215/128/184	Virtex-7	305/1001/558/0	245	5	-
This Work	FrodoKEM-640	2.8/16/12	Virtex-7	71/203/106/0	292	1	-
	FrodoKEM-976	2.3/16/10		65/179/92/0	318	1	
	FrodoKEM-1344	1.4/15/6		41/109/80/0	351	3	
This Work	SEAL-128	3.19/128/41	Virtex-7	654/2347/581/0	152	2	-
	SEAL-192	3.19/192/51		993/3620/843/0	122	2	
	SEAL-256	3.19/256/60		845/4845/1103/0	102	2	
This Work	FALCON-I	2/53/18	Virtex-7	184/627/248/0	227	2	-
	FALCON-II	$\sqrt{5}$/200/37		626 /2142/849/0	116	2	
HW [17]	-	4.41/112/55	Spartan-6	122/426/123/**1**	102	8	5.01×
HW [18]	-	4.41/112/55	Spartan-6	150/463/45/0	80	30	15.69×
This Work[a]	-	4.41/112/55	Virtex-7	298/970/549/0	263	3	-

Aydin Aysu et al. "Efficient, flexible, and constant-time gaussian sampling hardware for lattice cryptography." *IEEE Transactions on Computers* 71, no. 8 (2021): 1810-1823.

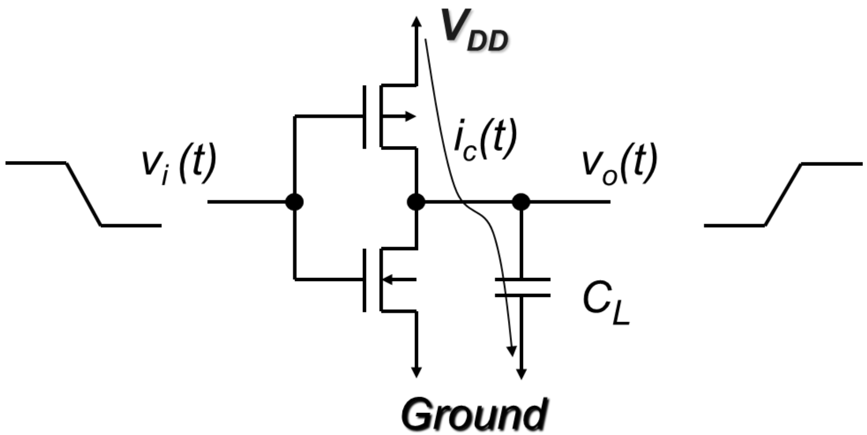
Implementation Security

New applications (e.g. IoT) expose hardware to direct physical attacks / tampering: breaks crypto / key stolen



Physical Side-Channel Analysis

This talk: Power and EM



$$P_{cpu} = P_{dyn} + P_{stat} = P_{tran} + P_{sc} + P_{leak}$$
$$P_{tran} = C_L V_{DD}^2 \bullet f \bullet P_{0\rightarrow 1}$$

Fundamental property of CMOS:

+ More practical (low-cost) than optical leakage
+ More precise than thermal leakage

Side-Channel Security

Physical source: power, EM, acoustic, photonic, thermal, …

Digital source: time, micro-architectural state, memory patterns, …

Differential Power Analysis

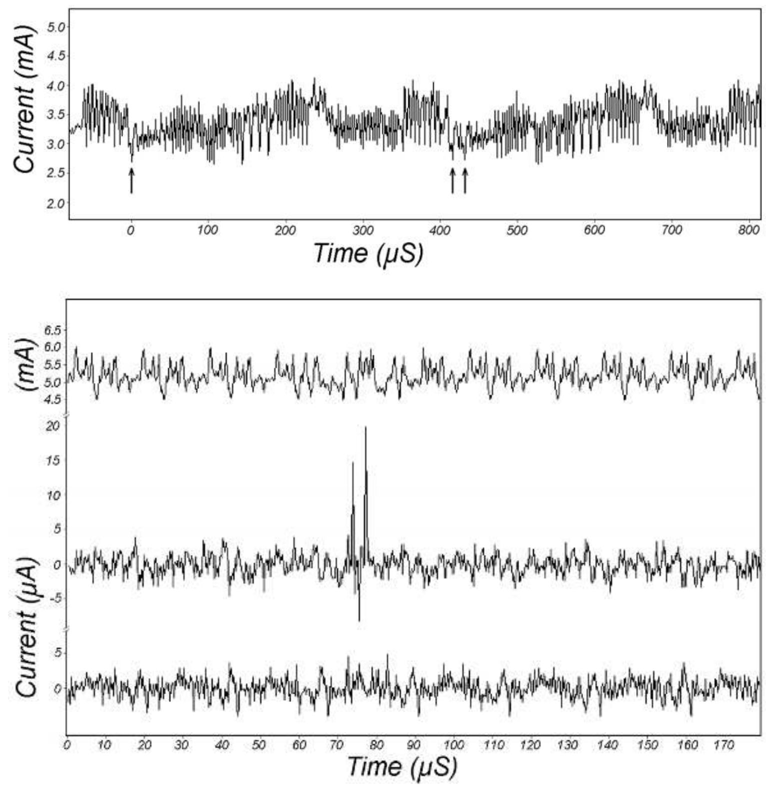
Paul Kocher, Joshua Jaffe, and Benjamin Jun

Cryptography Research, Inc.
607 Market Street, 5th Floor
San Francisco, CA 94105, USA.
http://www.cryptography.com
E-mail: {paul,josh,ben}@cryptography.com.

Abstract. Cryptosystem designers frequently assume that secrets will be manipulated in closed, reliable computing environments. Unfortunately, actual computers and microchips leak information about the operations they process. This paper examines specific methods for analyzing power consumption measurements to find secret keys from tamper resistant devices. We also discuss approaches for building cryptosystems that can operate securely in existing hardware that leaks information.

Keywords: differential power analysis, DPA, SPA, cryptanalysis, DES

CRYPTO'99*



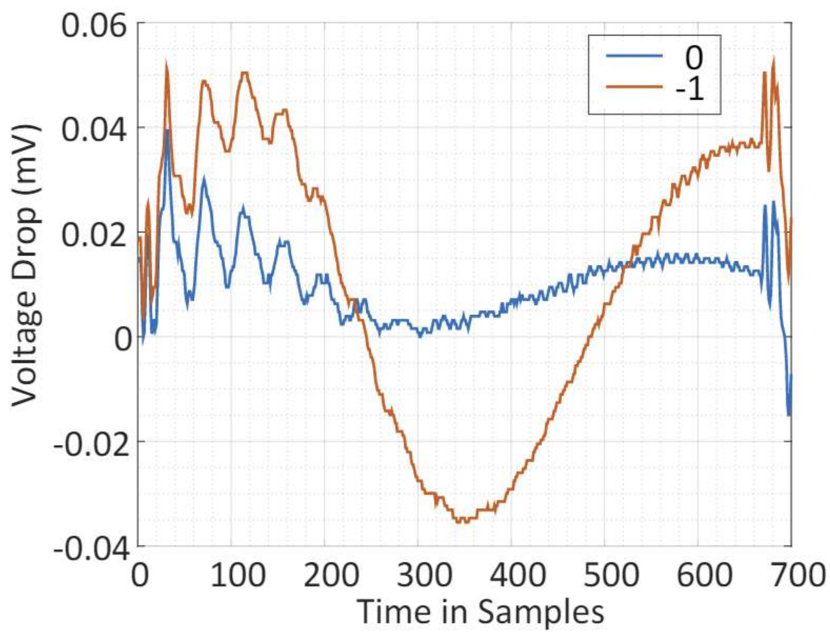
*Omitting TEMPEST for simplicity

FALCON's Side-Channel Vulnerability

Key generation sub-routine leaks secret key bit values

```
static inline uint32_t
mq_conv_small(int x)
{
    uint32_t y;
    y = (uint32_t)x;
    y += Q & -(y >> 31);
    return y;
}
```

00000000 FFFFFFFF



NTRU and NTRU Prime Side-Channel Vulnerability

Listing 1. NTRU Sorting Reference Implementation

```
void crypto_sort_int32(
        int32 *array, size_t n)
{
  ...
  for (p = top;p >= 1;p >>= 1){
    i = 0;
    while (i + 2 * p <= n){
      for (j = i;j < i + p;++j){
       int32_MINMAX(x[j],x[j+p]);
      }
      i += 2 * p;
    }
  ...
}
```

```
#define int32_MINMAX(a,b) \
do { \
  int32_t ab = (b) ^ (a); \
  int32_t c = \
       (int32_t)((int64_t)(b)
       - (int64_t)(a)); \
  c ^= ab & (c ^ (b)); \
  c >>= 31; \
  c &= ab; \
  (a) ^= c; \
  (b) ^= c; \
} while(0)
```

0x0000_0000 (0)

0xFFFF_FFFF (-1)

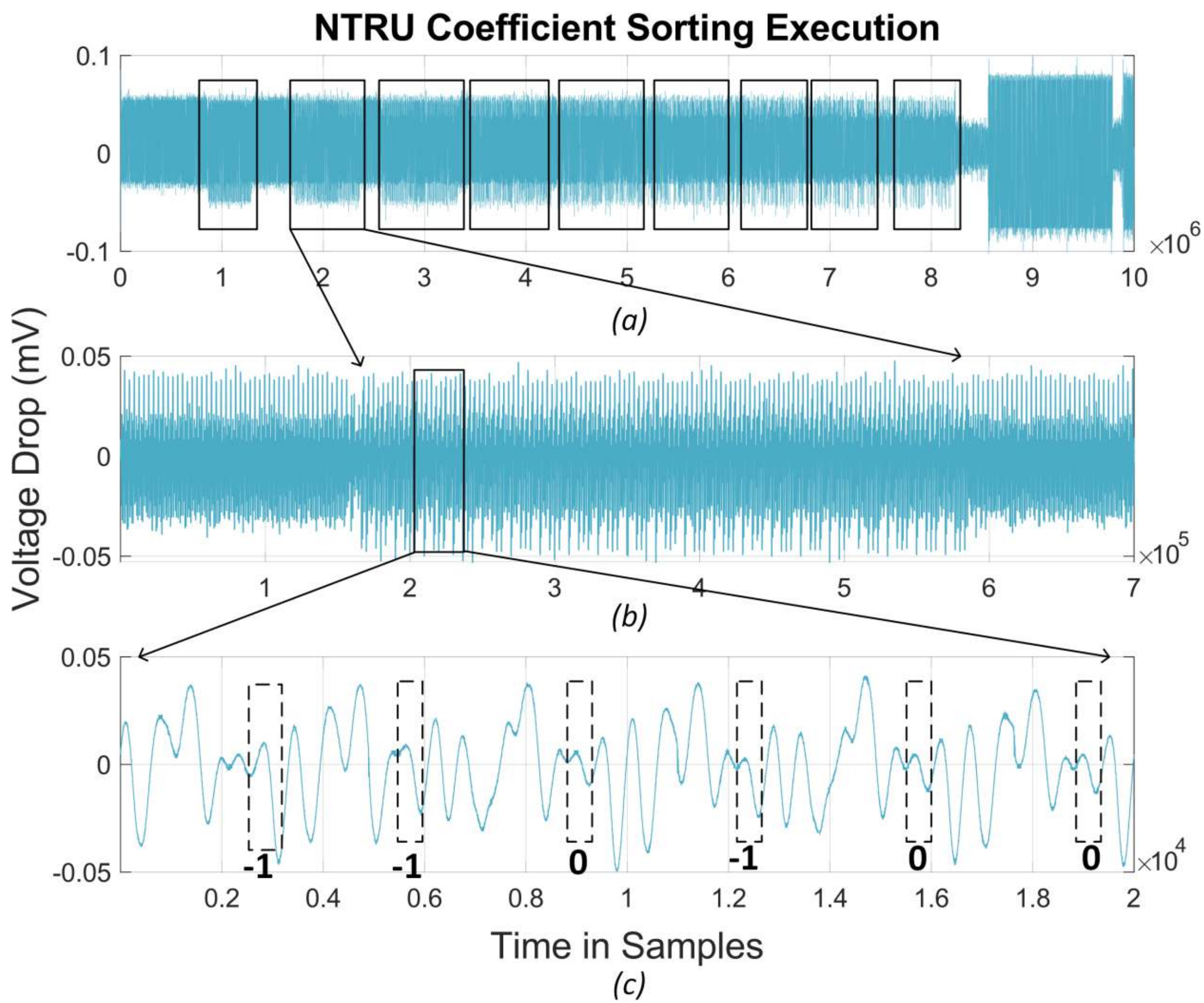
Karabulut, Emre, Erdem Alkim, and Aydin Aysu. "Single-trace side-channel attacks on ω-small polynomial sampling: with applications to NTRU, NTRU prime, and crystals-dilithium." In *2021 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 35-45. IEEE, 2021.

SamplePerm Full Power Trace and Extraction

```
#define int32_MINMAX(a,b) \
do { \
  int32_t ab = (b) ^ (a); \
  int32_t c = \
      (int32_t)((int64_t)(b)
      - (int64_t)(a)); \
  c ^= ab & (c ^ (b)); \
  c >>= 31; \
  c &= ab; \
  (a) ^= c; \
  (b) ^= c; \
} while(0)
```

0x0000_0000 (0)

0xFFFF_FFFF (-1)

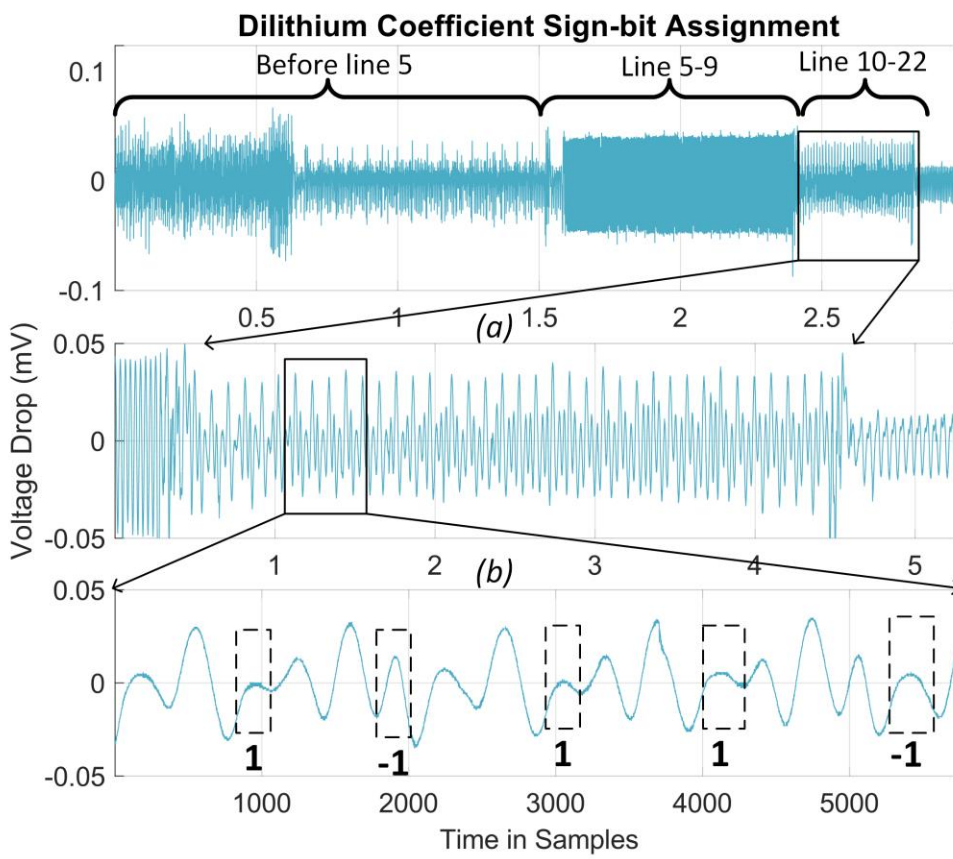


Karabulut, Emre, Erdem Alkim, and Aydin Aysu. "Single-trace side-channel attacks on ω-small polynomial sampling: with applications to NTRU, NTRU prime, and crystals-dilithium." In *2021 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 35-45. IEEE, 2021.

Dilithium Sampling Leakage

Listing 3. Dilithium Polynomial Generation Reference Implementation

```
1  void poly_challenge(poly *c,
2                   const uint8_t seed[SEEDBYTES])
3  {
4    ...
5    for(i = 0; i < 8; ++i)
6      signs |= (uint64_t)buf[i] << 8*i;
7    pos = 8;
8    for(i = 0; i < N; ++i)
9      c->coeffs[i] = 0;
10   for(i = N-TAU; i < N; ++i) {
11     do {
12       if(pos >= SHAKE256_RATE) {
13         shake256_squeezeblocks(buf, 1,
14                                          &state);
15         pos = 0;
16       }
17       b = buf[pos++];
18     } while(b > i);
19     c->coeffs[i] = c->coeffs[b];
20     c->coeffs[b] = 1 - 2*(signs & 1);
21     signs >>= 1;
22   }
23 }
```



(c)

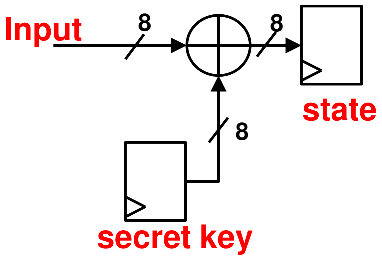
Karabulut, Emre, Erdem Alkim, and Aydin Aysu. "Single-trace side-channel attacks on ω-small polynomial sampling: with applications to NTRU, NTRU prime, and crystals-dilithium." In *2021 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 35-45. IEEE, 2021.

Requirements For A Differential Side-Channel Attack

An intermediate computation:

1) that combines a known value and a secret key and

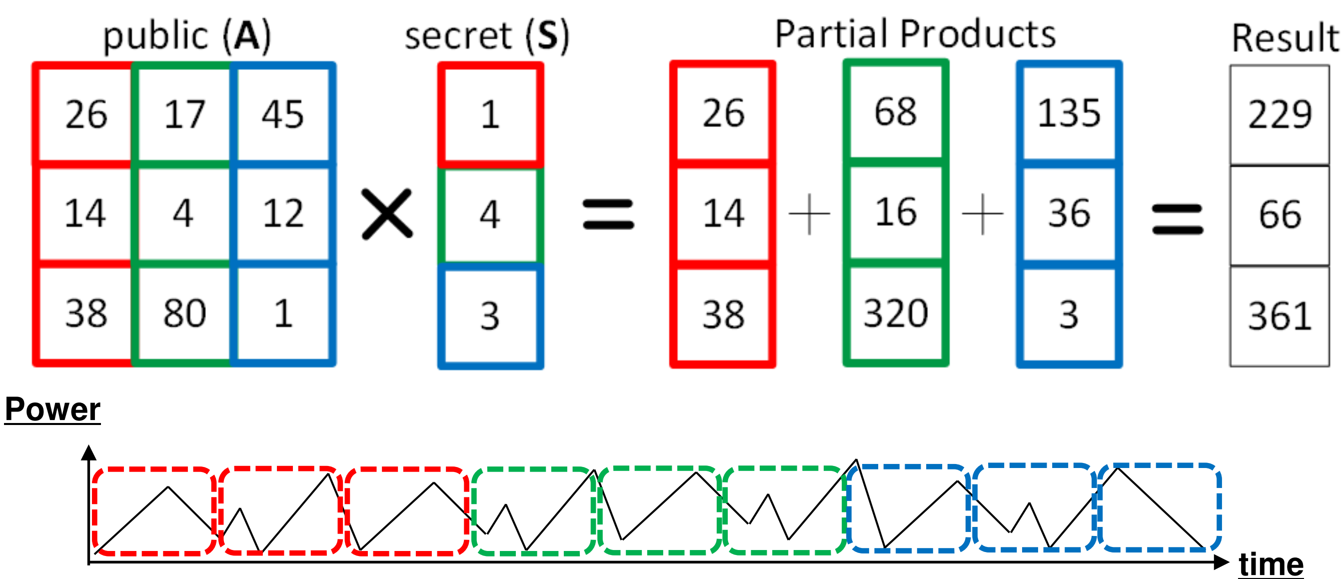
2) the known value varies (*i.e.*, not fixed)



	Key Hypothesis							
	Key=00		Key=01		---------	Key=ff		Power
Input	state	P_m	state	P_m		state	P_m	(µW)
I_1=01	01	1	00	0		fe	7	
I_2=0f	0f	4	0e	3	---------	f0	4	
⋮	⋮	⋮	⋮	⋮		⋮	⋮	⋮
I_{10000}=f1	f1	5	f0	4	---------	0e	3	

Single-Trace Differential Attacks on FrodoKEM Matrix Multiplication

- ❑ Attacker limited to a single power measurement trace
- ❑ Matrix multiplication has "multiple" intermediate computations on the same secret
  - ❑ Up to 1344 distinct computations on the same secret (S) coefficient
  - ❑ Attack splits measurements into "sub-traces" for profiling and test



Attacking the FALCON Signatures with Differential Power Analysis

Algorithm 2 FALCON Signature Generation Algorithm [5]

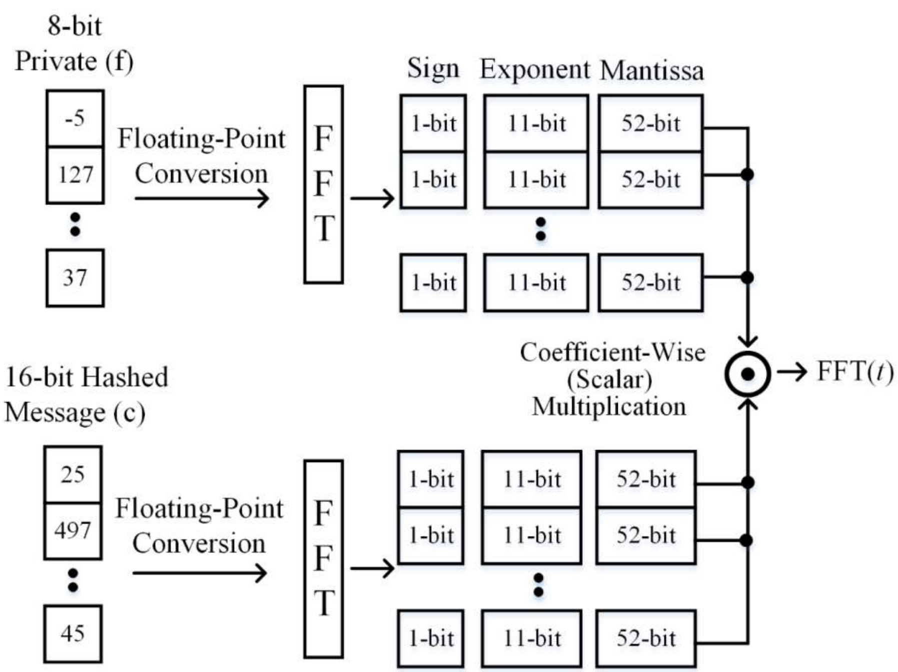
Input: a message m, a secret key sk, a bound β^2
Output: a signature sig of m

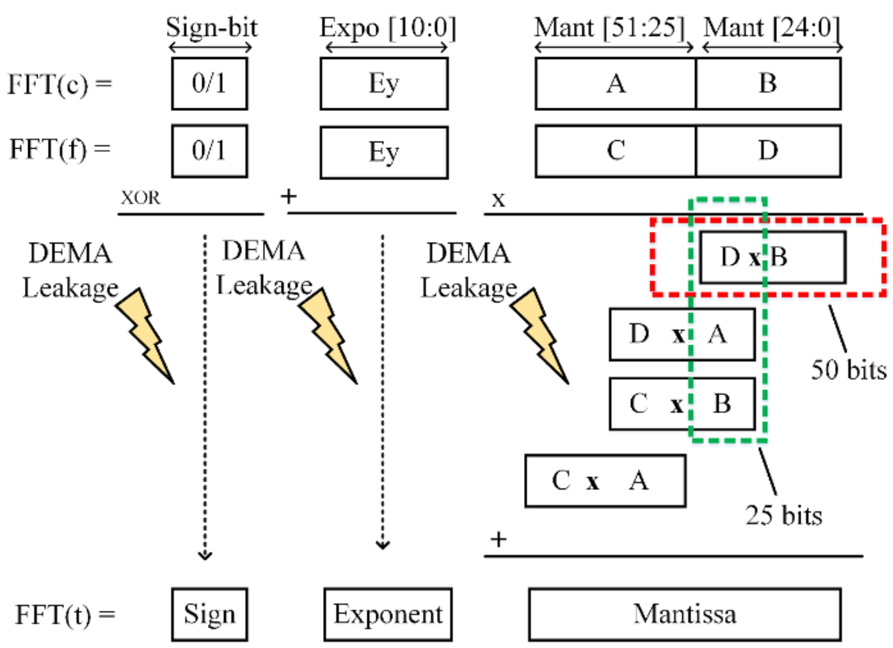
$r \leftarrow \boxed{\{0,1\}^{320} \text{ uniformly}}$
$c \leftarrow \boxed{\text{HashToPoint } (r||m)}$
$t \leftarrow (\frac{-1}{q} FFT(c) \odot FFT(F), \frac{1}{q} FFT\boxed{(c)} \odot FFT\boxed{(f)})$
do ▷ $\odot$ represents FFT multiplication
  do
    $z \leftarrow \text{ffSampling } (t, T)$
    $\mathbf{s} \leftarrow (t - z) \begin{bmatrix} FFT(g) & -FFT(f) \\ FFT(G) & -FFT(F) \end{bmatrix}$
  while $s^2 > \lfloor \beta^2 \rfloor$
  $(s_1, s_2) \leftarrow invFFT(s)$
  $s \leftarrow \text{Compress}(s_2, 8 \cdot sbytelen - 328)$
while $s = \perp$
return $sig = (r, s)$

- NTRU equation:
  $fG - gF = q$
- Public Key:
  $h = gf^{-1}$
- If we know either polynomial 'g' or 'f', we can recover the other secret polynomial
- **Attack target:** Multiplication of known polynomial 'c' and secret polynomial 'f'

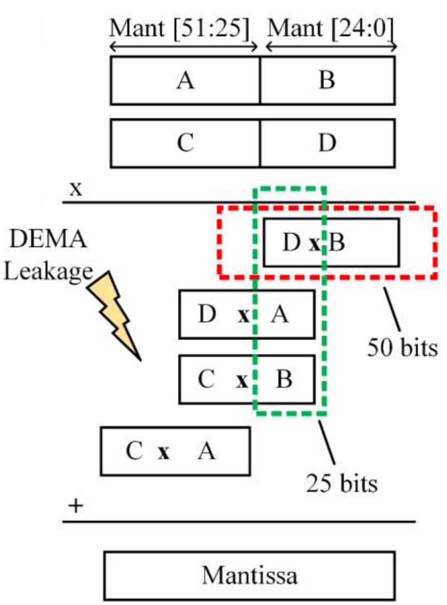
FALCON FFT and Multiplication

Secret coefficients of f can be recovered by targeting the FFT-domain multiplication





Challenge of Attacking Multiplication



B (known) 6 vs 6

D (guessed) 2 . 4

×

12 24 → Same Hamming Weight

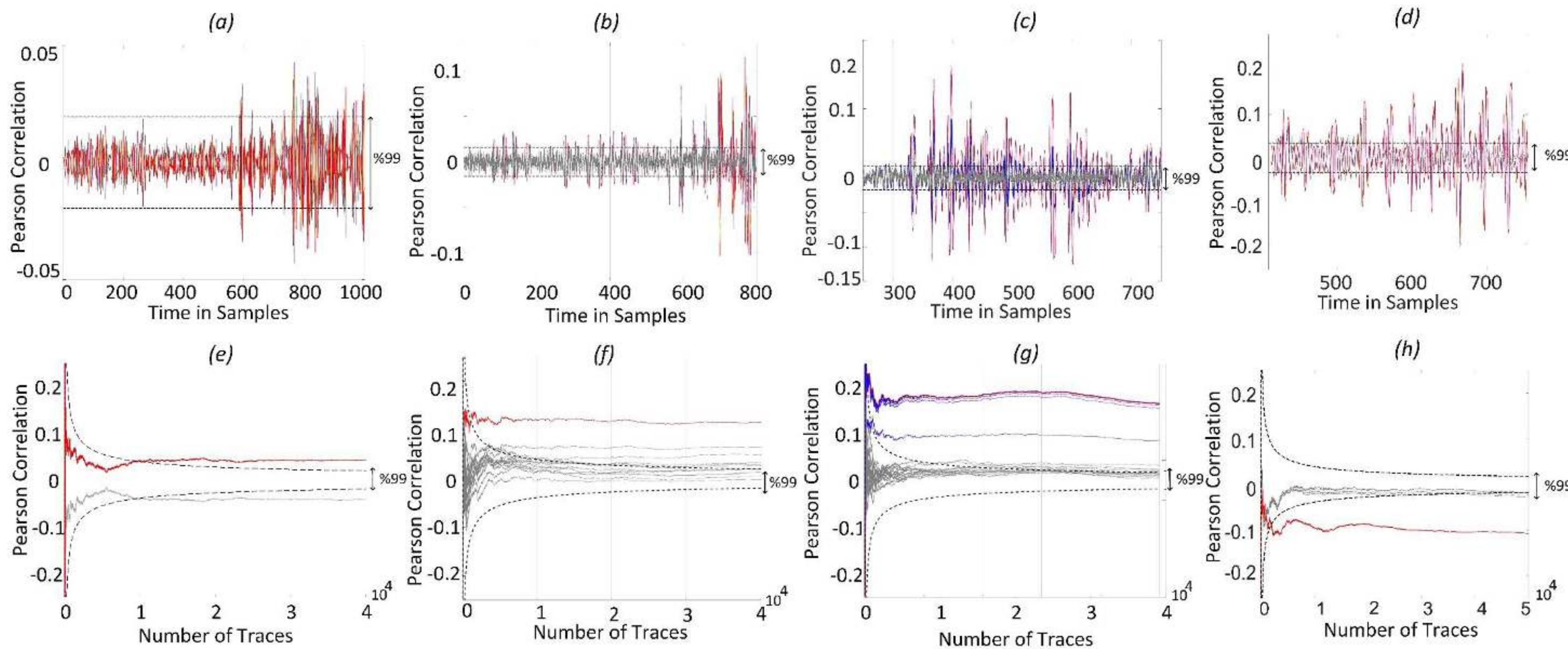
+ 5 + 5

17 29 → Different Hamming Weights

Additions remove false positives: apply extend-and-prune!

Evaluation Results

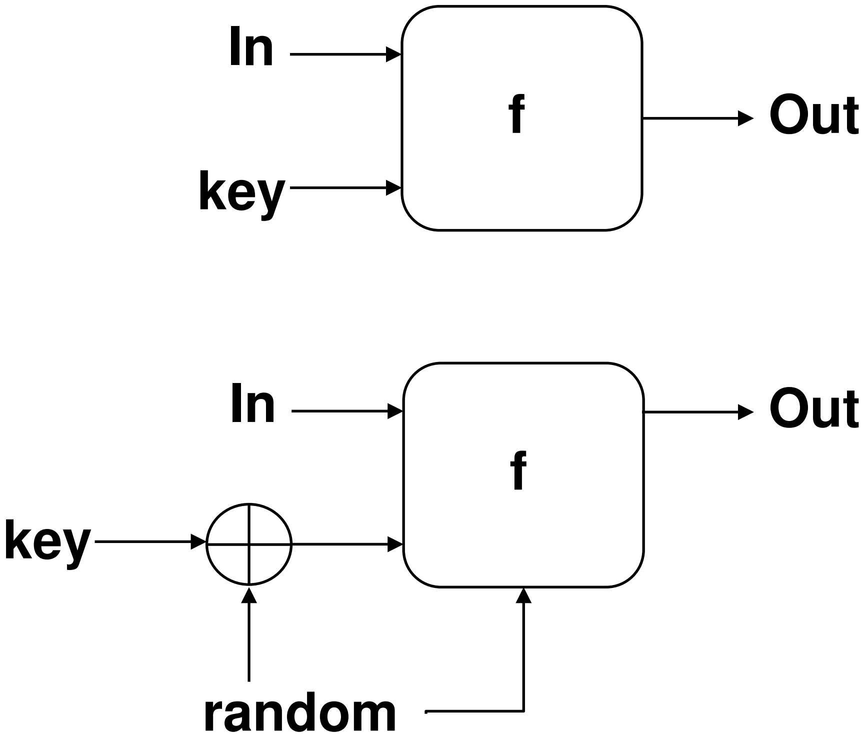
1k measurements can extract sign, 100 traces can extract exponent and mantissa



Attacking the Sign-bit | *Attacking the Exponent* | *Attacking the Mantissa Multiplication* | *Attacking the Mantissa Addition*

Side-Channel Security

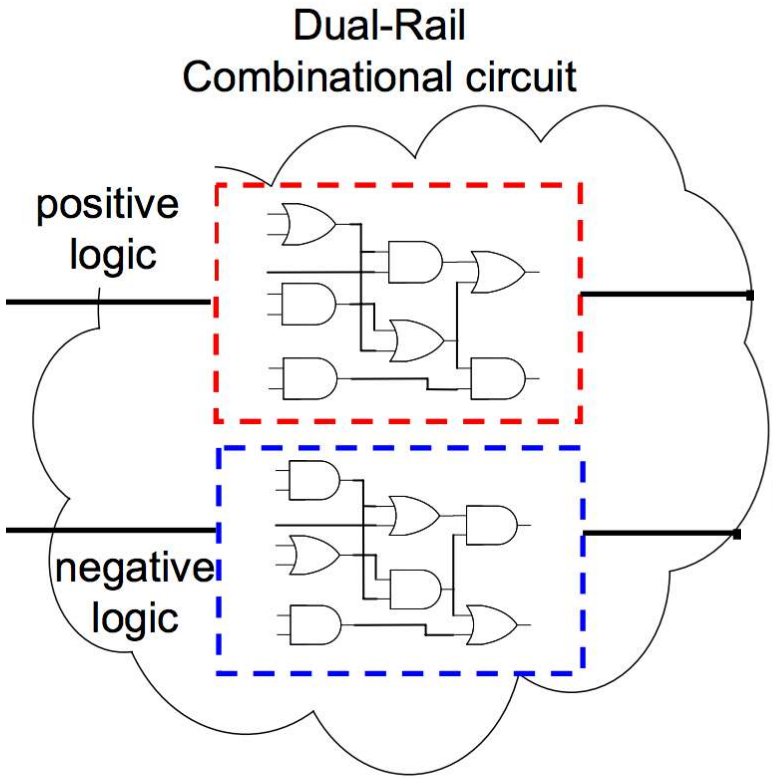
Masking



Randomize "intermediate" computations

+ Provably secure
- Needs tuning for each ***f***

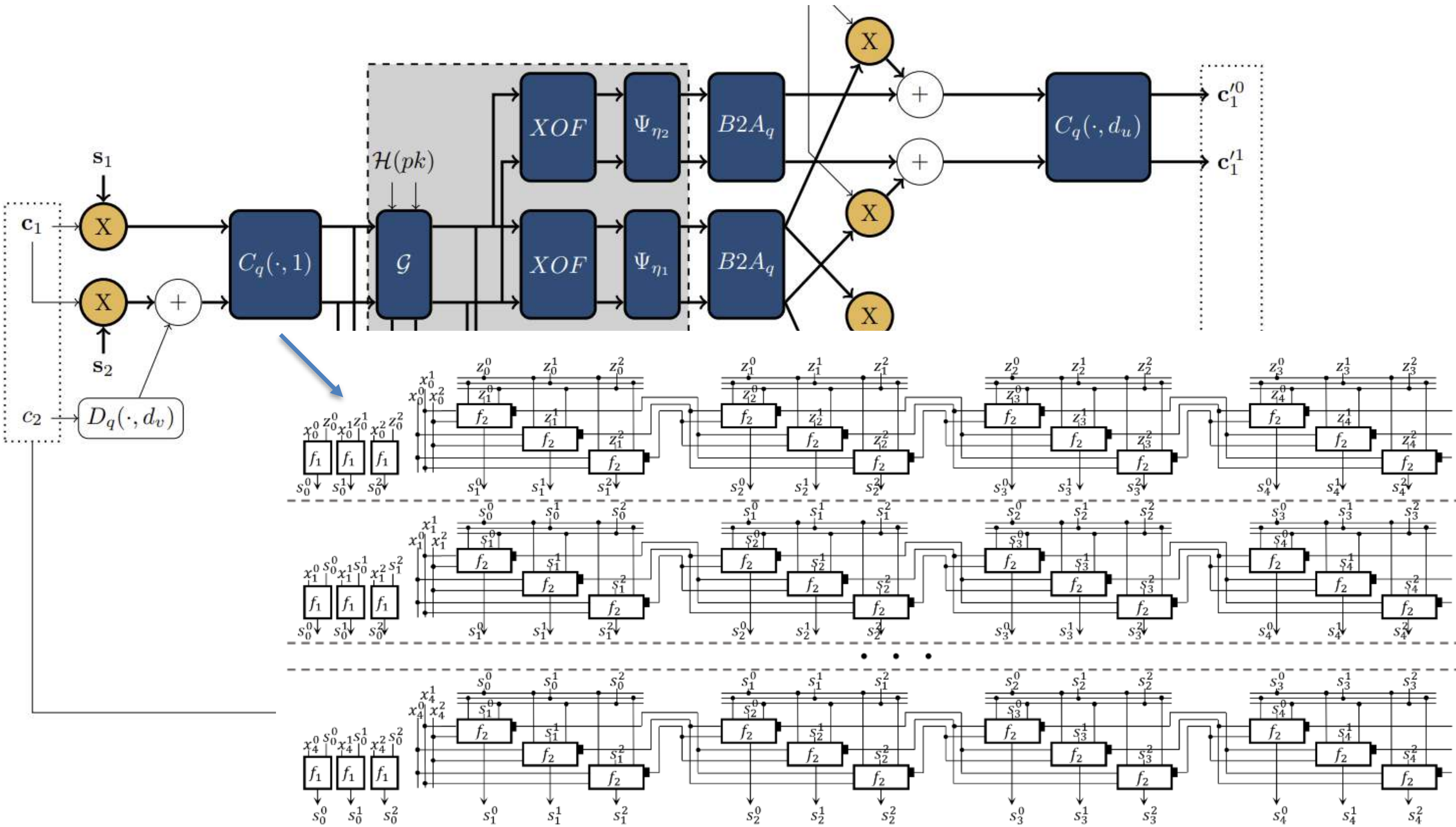
Hiding



Design constant power circuits

+ Automation friendly
- Patented* and may leak

Masking Cryptographic Hardware Is Hard!



Fritzmann, Tim, et al. "Masked accelerators and instruction set extensions for post-quantum cryptography." *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2022.1 (2021): 414-460.

Three Takeaways

1. Quantum-secure cryptography is unavoidable

2. (Lattice-based) quantum-secure cryptography is <u>fundamentally different</u>

3. Need new hardware designs:
   - Optimize components
   - Design full system and explore trade-offs & design space
   - Support hybrid schemes
   - Add “implementation” security

Questions

Research Sponsors

SRC® Semiconductor Research Corporation

NSF

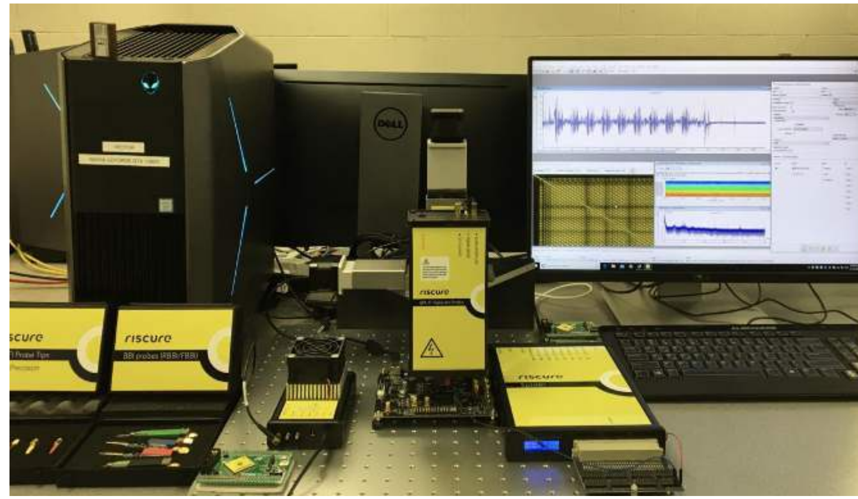
CAEML CENTER FOR ADVANCED ELECTRONICS THROUGH MACHINE LEARNING

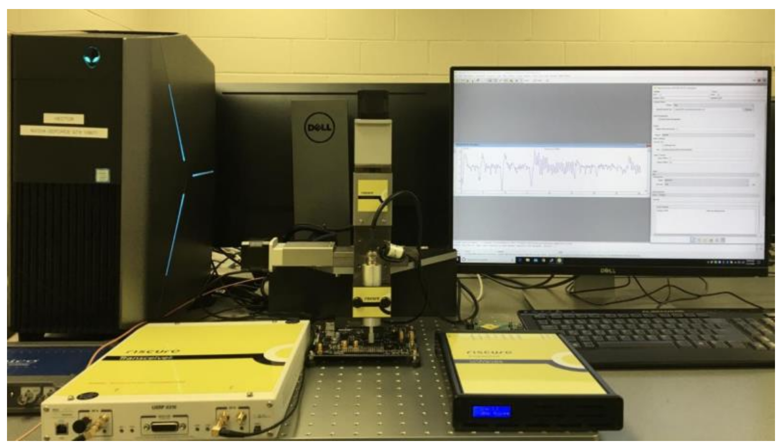
Office of Naval Research USN ONR Science & Technology

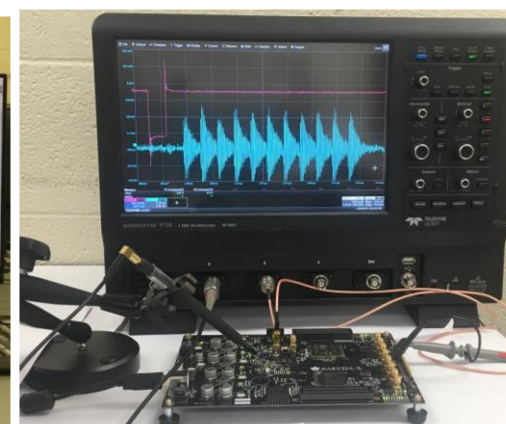
Google

riscure

Attack Setups







Interested? Email: **aaysu@ncsu.edu**

ChatGPT Answers

AA

When should we start using post-quantum cryptography?

It is recommended to start using post-quantum cryptography as soon as possible, even though the immediate threat of a quantum computer breaking current encryption methods is low. The reason for this is that it can take a long time to phase in a new cryptosystem and make sure it is secure and properly implemented. Additionally, as quantum computers continue to improve, the threat of them breaking current encryption methods will increase, so it is better to start preparing for this eventuality now.

It is important to note that post-quantum cryptography should not be used alone, but should be used in conjunction with currently used cryptographic methods. This will provide an extra layer of security, making it more difficult for an attacker to break the encryption even if they have access to a quantum computer.

It is also important to use post-quantum cryptography for sensitive information that needs to be protected for a long period of time, such as financial information, personal identification, and classified data.