

Quantum Computing, Cryptography, Cryptanalysis, and Codes

Classical and Quantum Computational Complexity: Theory and Practice

13 Apr 2026, Jain University, Bangalore

C.E.Veni Madhavan

Department of CSA, Indian Institute of Science, Bangalore
Centre for Cyber Security, Amrita Vishwa Vidyapeetham, Coimbatore
Department of CSE, Indian Institute of Technology, Bhilai

April 12, 2026

- 27-28, March 2024, IISc, Bangalore
- 1 - 4 May 2024, Amrita Vishwa Vidyapeetham, Coimbatore
- 25 Nov 2024, GITAM University, Vishakkapatnam
- 19 Mar 2025, CDAC Bangalore
- 4 September 2025, PKIA 25, CDAC-CCA, Bangalore
- 10-12, Nov 2025, Amrita Vishwa Vidyapeetham, Coimbatore
- 22-23, Jan 2026, CCS, CSE, Indian Institute of Technology, Hyderabad
- 05-07, Feb 2026, CCS, Amrita Vishwa Vidyapeetham, Coimbatore
- 19-20, Feb 2026, CSE, Indian Institute of Technology, Bhilai
- 18-20, Mar 2026, Scientific Analysis Group, New Delhi
- 13 , Apr 2026, Jain University, Bangalore

- *Public key Cryptography: Theory and Applications*, Abhijit Das, C.E.Veni Madhavan, Pearson Education, New Delhi, 2009.
- *Quantum computational and quantum information*, Nielsen, Chuang, Prentice Hall, New Delhi, 2000.

$$\text{co} - NP \subsetneq P \subsetneq NP$$

$$P \subset BPP \subset NISQ \subset BQP$$

Contents

- ① Mathematical Foundations: *Objects, Undercurrents* 1930-
- ② Cryptography, Cryptanalysis: *Turing, Shannon, Classical, Modern Algorithms* 1950-2015
- ③ High Performance Computing Trends 2015-2022
- ④ Quantum Computing 2017-2022
 - ① Investment, & Institutional Trends
 - ② Scientific, & Technological Trends
 - ③ Algorithmic Complexity Strands
 - ④ Physics and Engineering Strands
- ⑤ Post-Quantum-Public-Key-Cryptography 2019-2023
 - 5.1 codes: rank code - McEc; RQC; HQC BIKE-QCMDPC;
 - 5.2 lattices: - pqNTRU (FALCON); CRYSTAL (Kyber, Dilithium)
 - 5.3 curves: isogeny - SIKE, 5.4 equations: MVQ - RAINBOW
 - 5.5 congruences: ring lwe - NewHope 5.6 sequences: tree hash-SPHINCS
- ⑥ Quantum algorithms - IFP, DLP, ECDLP 1994-2023
- ⑦ Lattice Geometry based Classical and Quantum Factoring 1994-2023
- ⑧ Super-singular Isogeny based DH, KE of PQC 2011-2023

1.1 Mathematical Foundations : objects

- **Objects** : $\mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}, \mathbb{F}, \mathbb{K}, \Sigma$; *polynomials, words, sets, sequences*
- **Algebraic Structures** : *groups, rings, fields, modules, spaces*
- **Combinatorial Structures** : *graphs, matroids, cycles, braids, posets*
- **Compound Structures** : *codes, ciphers, curves, lattices, sequences*
- **Mappings** : *endo/homo/iso morphisms; isogenies; homo/iso nymys*
- **Relations** : *equations, congruences, memberships, containments*
- **Invariants** : *orders, symmetries, equivalences, transforms, norms*
- **Algorithms** : *add/sub, mul/div, mod, exp, log, equations-disc./cont.*
- **Realizations** : *hw/sw; wire-line, wireless; devices, systems, interfaces*
- **Complexities** : *structural, descriptive, prescriptive, computational*
- **Analyses** : *size, space, time, m/c cycles, errors, asymptotics, averages*

1.2 Mathematical Foundations : Undercurrents

- *Code Breaking* [Shannon, Turing, Friedmann, Kahn, Shamir ...]
- *Numbers with no large prime factors* [Ramanujan]
- *A Mathematician's Apology* [G.H.Hardy]
- *Unreasonable Effectiveness of Mathematics* [E.Wigner]
- *Mathematical Thinking versus Algorithmic Thinking* [D.E.Knuth]
- *Tale of Two Sieves* [Pomerance]
- *xedni Calculus* [Silverman]
- *The Uneasy Relationship between Math. and Crypto.* [Koblitz]
- *A Riddle Wrapped in Enigma* [Koblitz and Menezes]
- *Low-resource Quantum Factoring* [D.Bernstein et al]
- *Measure concentration, Weak Pinsker Conjecture* [T.Austin]
- *Mathematics and Computer Science - the Interplay* [cevm]
- *ugadpac-gotgor-amplify* [cevm]

2. Cryptography, Cryptanalysis and Computations

We present a few slides on the computational complexity of *all* forms of *cryptographic* and *cryptanalysis* techniques. The focus is on the *harder* problem of cryptanalysis. We attempt to capture the broad spectrum of attack methodologies. We try to provide upper-bound estimates on machine cycles for key recovery.

Typical 3 GHz, linux, C Benchmarks:

- Stream Ciphers ($\simeq 1.5$ Gbits/sec) :
LFSR, non-linear FSR, FISH, PIKE, A5 ...
- Block Ciphers ($\simeq 300$ Mbits/sec) :
DES, IDEA, BLOWFISH, RC5 (64 bit); RC6, TWOFISH, MARS, RIJNDAEL, SERPENT (AES-128 bit)
- Public Key Ciphers ($\simeq 20$ Kbits/sec) :
RSA, ElGamal ($\mathbf{F}_p; \mathbf{F}_q, q = 2^n, p^n$), Elliptic Curve ($\mathbf{E}(\mathbf{F}_q)$);
Chor-Rivest, NTRU ...
- Digital Signatures
(generation $\simeq 20$ Kbits/sec, verification $\simeq 1.2$ Mbits/sec) :
RSA, ElGamal ($\mathbf{F}_p; \mathbf{F}_q, q = 2^n, p^n$), Elliptic Curve ($\mathbf{E}(\mathbf{F}_q)$); blind,
undeniable, group ...
- Mitsubishi Smart card M16C - 16bit, 10MHz, 64KB ROM, 4KB RAM
400 msec RSA Verification (1024 bit modulus, $e = 65537$)

- ① **Integer Factoring Problems (IFP)** Let N be an integer with $N = p * q$ for prime integers, p, q . Given N find the factors.
- ② **Discrete Logarithm Problems (DLP)** Let G be a group. The groups to be considered are (i) the multiplicative group of the finite field $\mathbf{F}_q$, for q an odd prime or $q = 2^m$, (ii) the additive group of points on an elliptic curve over a finite field $E(\mathbf{F}_q)$. Let g be a fixed, distinguished element (e.g., a generator of a cyclic group or an element of large order) of G and let $a = g^x$ for some x . Given g, a in G determine x .
- ③ **Statistical Analysis Problems (SAP) - cryptanalysis** Given the cipher-text $c = \langle c_0, \dots, c_n \rangle, c_j \in \{0, 1\}$ output of (i) a stream cipher or (ii) a block cipher, determine the corresponding (i) plain-text $p = \langle p_0, \dots, p_n \rangle, p_j \in \{0, 1\}$ or, (ii) symmetric key $k = \langle k_0, \dots, k_m \rangle, k_j \in \{0, 1\}$, *under various cryptanalytic scenarios*.

Cryptanalysis: Techniques

- **Historical Perspectives** (use PT idiosyncrasies. together with machine or algorithm idiosyncrasies to get cribs and then leverage the advantages - ENIGMA, HAGELIN, SIGABA)
- **Modern Perspectives**
 - *Malleability attacks* (modify the CT one bit at a time in the *right place* to change the semantics of the decrypted PT!)
 - *Divide-and Conquer attacks* (consider separately, the odd and even bits)
 - *Meet-in-the-Middle attacks* (given a known PT, CT pair analyses variations and correlations simultaneously from both input and output)
 - *Correlation attacks* (identifying statistical biases in the key stream from the structure of the nonlinear filter function and the clocking pattern)
 - *guess-and determine attacks* (only a few bits are tapped as input to the filter function or combiner or clocking function)
 - *Differential cryptanalysis* (build input differences and output difference correlation through a large number of challenges!)
 - *Linear cryptanalysis* (build linear relationships based on input differences and output difference)
 - *Algebraic attacks* (set up nonlinear equation, perform linearization, solve using linear algebraic techniques; Grobner basis; SAT solvers)

• My Perspectives:

- *ugadpac-amplify-coa* (guess and determine PT, predict-and correct CT, using approximate multi-point local interpolation and filtration strategy for cipher-text-only-attack)
- hybrid-classical-quantum algorithms: design, analysis for IFP, DLP, ECDLP
- hybrid-classical-quantum computations: scaled implementations of algorithms for IFP, DLP, ECDLP
- hybrid-classical-quantum algorithms: design, analysis for SVP, NVP, LWE
- hybrid-classical-quantum computations: scaled implementations of SVP, NVP, LWE
- hybrid-classical-quantum algorithms: design and analysis for decoding specific structured codes, arbitrary codes
- hybrid-classical-quantum computations: scaled implementations of decoding specific structured codes, arbitrary codes
- hybrid-classical-quantum algorithms: symmetric-key ciphers[hers
- hybrid-classical-quantum computations: for scaled implementations of specific structured codes, arbitrary codes

- Stream Ciphers :
linear complexity profile, correlations, mul. var. poly. eqns ...
- Block Ciphers :
differential, linear, Mod n attacks, algebraic attacks ...
- Public Key Ciphers integer factorization, discrete logarithms in groups, lattice short vectors, modular square roots ...
- side channel attacks - timing attacks, power analysis ...
- 1 Day = 86400 $\sim 2^{16}$ seconds; 1 Year = 2^{25} seconds,
- (assuming 1 single precision int/float mul instruction = 1 cycle);
1 MIPS/ 1 Mflops Year = 2^{45} cycles ;
1 BIPS/ 1 Gflops Year = 2^{55} cycles ;
1 TIPS/ 1 Tflops Year = 2^{65} cycles ;
1 PIPS/ 1 Pflops Year = 2^{75} cycles ;
1 EIPS/ 1 Eflops Year = 2^{85} cycles ; (**cryptanalysis limit**)

Typical Work factors for known *best* heuristics: Modern Ciphers

Integer factoring :

size (bits)	512	1024	2048
work (cycles)	$2^{\{64\}}$	$2^{\{86\}}$	$2^{\{116\}}$

Discrete logarithm in F_q

size (bits)	512	1024	2048
work (cycles)	$2^{\{60\}}$	$2^{\{80\}}$	$2^{\{100\}}$

Discrete logarithm in $E(F_q)$, $J(F_q)$

size (bits)	160	200	240
work (cycles)	$2^{\{70\}}$	$2^{\{90\}}$	$2^{\{120\}}$

DES(16 rounds) key 56 bits;

work (straight) : $2^{\{65\}}$ cycles

work (DC/LC) : $2^{\{55\}}$ cycles

AES (Rijndael- 10 rounds):key 128 bits; work: $\sim 2^{\{110\}}$ cycles

most stream ciphers key 128 bits; work: $\sim 2^{\{110\}}$ cycles

Typical Work factors for known *best* heuristics: Classical Ciphers

Transposition cipher

size (chars)	400	900	1600
work (cycles)	$2^{\{50\}}$	$2^{\{56\}}$	$2^{\{59\}}$

Poly-alphabetic Substitution Ciphers

classical block ciphers: key ~ 16 chars; work: $\sim 2^{\{20\}}$ cycles

Rotor Machines

classical stream ciphers: key ~ 16 chars; work: $\sim 2^{\{30\}}$ cycles

Enigma

classical stream ciphers: key ~ 16 chars; work: $\sim 2^{\{30\}}$ cycles

Hagelin Machines

classical stream ciphers: key ~ 32 chars; work: $\sim 2^{\{40\}}$ cycles

Typical work factors for hash chain based cryptanalysis,
cryptocurrency mining

An estimate of TMDTO attacks on password, passphrase recovery:
For a target of up to 12 characters (about 2^{70} to 2^{78} bits entropy) in real time of a few days, based on a rough design of rainbow tables consisting of 2^{40} chains of length 2^{30} each,

- (a) storage required: ~32 Petabytes,
- (b) pre-computing time: ~1 Pfyr (1 Petaflop machine for a year)
- (c) on-line cracking time: ~18 days on a 50 Tfyr,
- (d) capital cost of Rs. 5 crores, running cost of Rs. 10 lakhs/year.
- (e) TMT0, TMDTO, TMDCT0 attacks
- (f) proof-of-work computations for mining crypto-coins is about 2^{80}

Work Factors: Factoring RSA moduli

- 1995 RSA-130 : 432 : $\exp(1.93 * 6.69 * 3.19) = \exp(41.18) = 2^{59.41}$:
 $2^{59} < L(n, 1/3, 1.93) < 2^{60}$
- 1999 RSA-512 : 512 : $\exp(1.93 * 7.08 * 3.25) = \exp(44.10) = 2^{63.62}$:
 $2^{63} < L(n, 1/3, 1.93) < 2^{64}$
- 2003 RSA-576 : 576 : $\exp(1.93 * 7.36 * 3.30) = \exp(46.88) = 2^{67.6}$:
 $2^{67} < L(n, 1/3, 1.93) < 2^{68}$
- 2005 RSA-640 : 640 : $\exp(1.93 * 7.63 * 3.34) = \exp(49.18) = 2^{70.85}$:
 $2^{70} < L(n, 1/3, 1.93) < 2^{71}$
- 2010 RSA-768 : 768 : $\exp(1.93 * 8.11 * 3.40) = \exp(53.03) = 2^{76.51}$:
 $2^{76} < L(n, 1/3, 1.93) < 2^{77}$
- 1 ...
- 202? RSA-1024:1024 : $\exp(1.93 * 8.92 * 3.51) = \exp(60.42) = 2^{87.17}$:
 $2^{87} < L(n, 1/3, 1.93) < 2^{88}$
- 2 ...
- 20?? RSA-2048:2048 : $\exp(1.93 * 11.23 * 3.75) = \exp(81.28) = 2^{117.26}$:
 $2^{117} < L(n, 1/3, 1.93) < 2^{118}$

$L(n,c,e)=\exp\{c*(\ln n)^{(1/3)*(\ln(\ln (n)))^{(2/3)}\},$
 $c = 1.923, e=1/3$

size	exponent	u	practical bounds $T = 2^u$:
463	61.113070	54	(13000 hrs.@3GHz: $\sim 2^{(57)} > \sim 2^{(54)}$)
512	63.929363	56.3	(?)
576	67.375411	58.9	(?)
640	70.599666	62	(40 Opteron, 1yr: $\sim 40*3*2^{(30)}*2^{(25)}$) (Bonn, Nov'05)
704	73.636428	65.5	(?)($\sim 11.3*40 = 452$ Opteron yrs)
768	76.512120	69.3	(?)($\sim 13.93*452=6296$ Opteron yrs)
.	...	.	...
1024	87.176145	75 - 78	(1 Pfyr - 8 Pfyr) (!)(pre-quantum) (capital cost of Rs. 80 crores/Pflop [2014], running cost Rs. (June 2018, cost estimate : Rs.7 crores/Pflop) (Estimates of Lenstra et al [Asiacrypt 2003], Kleinjung [2017])
2048	117.263848	97 - 99	(1 TnEfy) (??? post-quantum)

intfact.pdf

2.1. Integer Factoring

Factor given integer n into its prime factors. Special interest $n = p * q$

① Trial Division

Divide n by all primes $p \leq \sqrt{n}$.

Complexity: $O(n / \log n)$; exponential in $\log n$.

② Group Order Methods

based on the *smoothness* of order of a *hidden group*

① $p - 1$ method

② $p + 1$ method

③ ρ -method and λ -method

④ *Elliptic Curve Method* (ECM)

③ Quadratic Congruence Methods

Basic idea by Fermat. Odd n can be written as $n = x^2 - y^2$.

Solving the congruence $X^2 \equiv Y^2 \pmod{n}$ with $X \not\equiv Y \pmod{n}$;

Two factors are $\gcd(X + Y, n)$ and $\gcd(X - Y, n)$.

① Dixon's Method

② Quadratic Sieve Method

③ Number Field Sieve Method

Integer Factoring - Group Order Methods

$p - 1$ Method by Pollard:

- 1 Based on Fermat's little Theorem
 $\forall p$, if $\gcd(a, p) = 1$ then $a^{p-1} \equiv 1 \pmod{p}$.
- 2 Integer N is B -smooth if $p \leq B$ for every prime $p|N$
e.g., 34496 is 11-smooth ($34496 = 2^6 * 7^2 * 11$).
- 3 Integer N is B -power smooth if $p^e \leq B$ for every prime with $p^e || N$.
- 4 This method succeeds if p (say) is B -power smooth.
 $\gcd(a^{B!} - 1, n) = p$.
e.g., $n = 443623$, $B = 11$, $B! = 39916800$ and
 $\gcd(3^{39916800} - 1, n) = 617$
- 5 Essentially, if this method works if the subgroup $\mathbb{Z}_p^*$ contained in $\mathbb{Z}_n$ has smooth order.

$p + 1$ Method:

- 1 Similar to $p - 1$ method but succeeds if $p + 1$ is smooth.
- 2 Much slower than $p - 1$

Integer Factoring - Group Order Methods

Elliptic Curve Method: (ECM)

- An *Elliptic Curve* **E** is a genus $g = 1$ *non-singular* curve given by

$$Y^2 = X^3 + aX + b \quad a, b \in \mathbb{F}$$

with $\Delta = -16(4a^3 + 27b^2) \neq 0$ over field $\mathbb{F}$

- Solutions of **E** over $\text{bfF} \times \mathbf{F}$, along with special point $\mathcal{O}$, forms additive abelian group; where given two points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$, $P \neq -Q$

- 1 Addition: $P \neq Q$, then $P + Q = (x_3, y_3)$ where,

$$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1} \right)^2 - x_1 - x_2; \quad (1)$$

$$y_3 = -y_1 + \left(\frac{y_2 - y_1}{x_2 - x_1} \right) (x_1 - x_2). \quad (2)$$

- 2 Doubling: $P = Q$, then $2P = (x_3, y_3)$ where,

$$x_3 = \left(\frac{3x_1^2 + a}{2y_1} \right)^2 - 2x_1; \quad (3)$$

$$y_3 = -y_1 + \left(\frac{3x_1^2 + a}{2y_1} \right) (x_1 - x_2). \quad (4)$$

- Above arithmetic is called *Chord-Tangent Law*.
- Hasse-Weil Theorem

$$p + 1 - 2\sqrt{\mathbf{F}} \leq E(\mathbf{F}) \leq p + 1 + 2\sqrt{\mathbf{F}}$$

Integer Factoring - Group Order Methods

Elliptic Curve Method: (ECM) (Contd.)

- Define $\mathbf{E}$ over $\mathbb{Z}_n$; doesn't form a group
- There are points P and Q such that $P + Q$ is not defined.
- Will get the factor of n when inverse doesn't exist while performing addition or doubling.
- OR factoring n succeeds if $\mathbf{E}(\mathbb{Z}_p)$ is smooth where $p|n$.
i.e., $(B!)P = \mathcal{O}$ in $\mathbb{Z}_p$ but $(B!)P \neq \mathcal{O}$ in $\mathbb{Z}_n$
- Has lots of advantages over other methods
 - Lots of curves are available to try over $\mathbb{Z}_n$
 - Parallelization is possible
 - Many optimizations are possible for *scalar multiplication* $((B!)P)$ computation.
- Best known among group order method.
- Complexity: Sub-exponential in size of the prime factor.

Quadratic Congruence Methods : QS, NFS, FFS

- 1 Build the *prime base* , $P = \{p_1, \dots, p_{\pi(v)}\}$, where $\pi(v)$ is the number of primes below v .
- 2 Pick a random $z, 1 \leq z \leq n - 1$ and let $w = z^2 \pmod{N}$.
- 3 Factorize w over the *prime base* P . Let $w = W \times \prod p_i^{\alpha_i}$.
- 4 if $W \neq 1$, then goto Step2 else accumulate sufficient number (say, $\pi(v) + 1$) of factorizations and store the vectors $\gamma = (\gamma_1, \dots, \gamma_{\pi(v)+1})$ where γ represents the parity vector of the exponents, $\gamma_i \equiv \alpha_i \pmod{2}$. Perform Gaussian elimination mod 2 on the parity vectors γ to get a zero vector.
- 5 if no nontrivial combination is generated in Step4 then goto Step2 else compute y as the product of prime powers obtained in Step4 and let x equal the product of the corresponding z .
- 6 if $x \not\equiv \pm y \pmod{N}$ then compute $\gcd(x \pm y, N)$ and *HALT* else delete the "first w and goto Step2

2. Integer Factoring - Number Field Sieve

Most cryptosystems are based on *Hardness* of

- Integer Factoring problem (IFP) i.e., given $n = pq$, factor n
- Discrete Logarithm Problem (given g , g^x , find x)
 - Finite fields and their extensions (DLP)
 - Algebraic curves: elliptic (ECDLP) & hyper-elliptic (HECDLP)
- Algorithms to solve them (best known):
 - Number Field Sieve for IFP
 - Index Calculus Methods (ICM) for DLP
 - Number Field Sieve (NFS)
 - Function Field Sieve (FFS) and variations
- Time complexity: $L_n(c, \epsilon) = O(\exp(c(\log n)^\epsilon (\log \log n)^{1-\epsilon}))$
if $\epsilon = 0$: linear, if $\epsilon = 1$: exponential in *size of the input*.
For above algorithm $\epsilon = 1/3$

Factoring Methods

Congruent Solution Methods

Given odd composite number $n (= pq)$, find factors.

- Fermat's Idea: Find $x, y \in \mathbb{Z}$, such that $x \not\equiv \pm y \pmod{n}$ and $x^2 - y^2 = n$.
- Dixon's Method: Randomly pick x , and check whether $x^2 - n$ is square.
- Kraithcik's Modification: Find $x, y \in \mathbb{Z}$, such that $x \not\equiv \pm y \pmod{n}$ and $x^2 \equiv y^2 \pmod{n}$.
- For random solution $x, y \in \mathbb{Z}$ such that $x^2 \equiv y^2 \pmod{n}$, we have $x \not\equiv \pm y \pmod{n}$ with probability $1/2$.
- Observation: find s_i, t_i for $i = 1, \dots, l$ s.t. $s_i \equiv t_i \pmod{n}$ for each i and both $\prod s_i$ and $\prod t_i$ are squares.
- Quadratic Sieve: $f(X) = X^2 - n$ & $m = \lfloor \sqrt{n} \rfloor$, with $X = m + i$, $i = 1, 2, 3, \dots$

Number Field Sieve for factoring

Theory: Basic Idea

- **Poly-select:** Let $f_1(X)$ and $f_2(X)$ be two irreducible polynomials over $\mathbb{Z}$, of degree d_1 and d_2 respectively.
- $\exists m$ s.t. $f_1(m) = f_2(m) = 0 \pmod{n}$.
- $f_1(\alpha_1) = f_2(\alpha_2) = 0$ for $\alpha_1, \alpha_2 \in \mathbb{C}$
- $\varphi_i : \alpha_i \rightarrow m$ is homomorphism from $\mathbb{Q}(\alpha_i)$ to $\mathbb{Z}_n$
- **Sieving & Linear Alg:** Find a set $S = \{(a, b) \in \mathbb{Z} \times \mathbb{Z} : \gcd(a, b) = 1\}$ s. t.

$$\prod_{(a,b) \in S} (a - \alpha_1 b) \text{ is a square in } \mathbb{Z}[\alpha_1], \text{ say } \beta_1^2,$$

$$\prod_{(a,b) \in S} (a - \alpha_2 b) \text{ is a square in } \mathbb{Z}[\alpha_2], \text{ say } \beta_2^2.$$

- Under the homomorphism φ_i they are congruent modulo n .
- **Square root:** Hence $\varphi(\beta_1)^2 \equiv \varphi(\beta_2)^2 \pmod{n}$.
- With probability 0.5, $1 < \gcd(\varphi(\beta_1) - \varphi(\beta_2), n) < n$.

Number Field Sieve (Contd.)

Practice

Polynomial Selection Difficult to find two polynomials of degree > 2 with a common root modulo n .

Let $f_1(X) = \sum_{i=0}^k c_i X^i$ and $f_2(X) = cX - d$.

$f_1(\alpha) = 0$ over $\mathbb{C}$, $f_1(m) \equiv f_2(m) \equiv 0 \pmod{n}$.

$\varphi : \alpha \rightarrow m$ the homomorphism from $\mathbb{Z}[\alpha]$ to $\mathbb{Z}_n$.

Sieving $S' = \{(a, b) \in \mathbb{Z} \times \mathbb{Z} : \gcd(a, b) = 1\}$ s. t. both $a + \alpha b$ (called *algebraic side*) and $ac - bd$ (called *rational side*) are *smooth* in $\mathbb{Z}[\alpha]$ and $\mathbb{Z}$ resp.

Each pair (a, b) is called *relation*.

Performed by *line sieving* and *lattice sieving* or judicious combination of both.

Number Field Sieve(Contd.)

Practice (Contd.)

Filtering and Linear Algebra Find a subset S of S' s. t.

$$\prod_{(a,b) \in S} (a - \alpha b) \text{ is a square in } \mathbb{Z}[\alpha], \text{ say } \beta^2,$$

$$\prod_{(a,b) \in S} (ac - bd) \text{ is a square in } \mathbb{Z}, \text{ say } \gamma^2.$$

Square Root Given β^2 compute $\beta \in \mathbb{Z}[\alpha]$.

In practice, $\varphi(\beta)$ computed without computing β

Easy to compute γ .

Final $\gcd(\varphi(\beta) - \gamma, n)$ gives a factor of n .

Involves Algebraic & Analytic interlude

Polynomials for RSA-512

The polynomials used for RSA-512:

- Non-linear polynomial

$$\begin{aligned}f_1(X) = & 119377138320 X^5 \\& -80168937284997582 X^4 \\& -66269852234118574445 X^3 \\& +11816848430079521880356852 X^2 \\& +7459661580071786443919743056 X \\& -40679843542362159361913708405064\end{aligned}$$

- Linear polynomial

$$f_2(X) = X - 39123079721168000771313449081$$

- Small factors of leading co-efficient

$$c_5 = 119377138320 = 2^4 3^2 5^1 11^2 19^1 41^1 1759^1$$

- Number of roots of $f_1(X)$ w.r.t small primes

21 roots for primes < 17 , 34 roots for primes in $[19, 97]$

Sieving

Case: Number Field Sieve for IFP

- $x \in \mathbb{Z}$ is y -smooth if $\forall p|x$ the $p \leq y$.
- $x \in \mathbb{Z}$ is (y, z, i) -semi-smooth if $p_1 > p_2 > \dots > p_r > 1$ are all the distinct prime divisors of x then $p_1 \leq z$ and $p_{i-1} < y$.
- Algebraic Norm of $a + \alpha b$: $F_1(a, b) = b^l f_1(a/b) = \sum_{i=0}^l c_i a^i b^{l-i}$ and
Rational Norm: $F_2(a, b) = b f_2(a/b) = ca - db$.
- Choose bounds B_0 and B_1
 $\mathcal{F}_r = \{p \in \mathbb{P} | p \leq B_0\}$, rational factor base.
 $\mathcal{F}_a = \{\wp \subset \mathbb{Z}[\alpha] | \text{Norm}(\wp) = p \leq B_0\}$, algebraic factor base
(first-degree prime ideals) .
In practice, look at (p, r) pairs with $p < B_0$ and $f_1(r) \equiv 0 \pmod{p}$
- Choose the Sieve Area $\mathcal{A} = [-A, A] \times [1, B]$ (based on the $f_1(X)$)

Sieving (Cont.)

Case: Number Field Sieve for IFP

- Perform Sieving over $\mathcal{A}$ by **Line Sieve** or **Lattice Sieve** or both.
- Collect large number of (a, b) (B_0, B_1, i) -semi-smooth relations, with $i = 1, 2, 3$

Line or Linear Sieve:

- Idea:
$$p \mid f(a, b) \implies p \mid f(a + p, b)$$
- Fix b
- $\forall p \in \mathcal{F}_r$ ($\wp \in \mathcal{F}_a$) update memory location corresponding to a for which p ($\wp$) divides $|a + mb|$ (norm of $a + \alpha b$).

Sieving

Parameters for Rational Sieve, Algebraic Sieve

- Bounds for factor base, large primes depend on size of input, which in turn determine the complexity of the method.
- **Rational Side**
 - Factor Base: 4000000 primes
 - Large Prime bounds:
Two large primes of size ≤ 28 bits
 - Rational residue size: 70 – 100 bits
- **Algebraic Side**
 - Factor Base: 4001293 first-degree prime ideals (p, r)
 - Large Prime bounds:
Three large primes (prime ideal norm) of size ≤ 30 bits
 - Norm size: 130 – 200 bits
- **Large Prime Factorization**
elliptic curve and other group order methods

Sieving

Line Sieve

- Sieving regions: $A - B$ plane

	+ve B		-ve B	
	start	end	start	end
A	1	2^{28}	1	2^{28}
B	1	$156928 (< 2^{18})$	-1	-156928

- Parallelization of blocks of lines:

Cluster: 280 nodes with each CPU is of 2.66 GHz

Block size: 16 B -lines per block

- Timings and yield:

Each block (16 B -lines, A -range $1 - 2^{28}$) took ~ 7 hrs¹.

Total timings for line sieve: 7.84 CPU years.

Total number of Relations (rows) = 7793906

Total number of prime ideals (p, r) = 14992106

Total number of primes = 7075731

Total number of columns = 22067837

¹Sieve within FB + Large Prime Factorization

Sieving

Line Sieve (Cont...)

- Data for analysis:

Norm sizes:

Algebraic Side: 130 – 200 bits

Rational Side: 70 – 100 bits

LPs:

bit size	Algebraic	Rational
27	2077244	1681700
28	2659676	2015018
29	3115697	0
30	3489351	0

k-tons:

1	2	3	4	5
13781089	3811519	1480066	769197	467431
6	7	8	9	≥ 10
310283	219856	163112	126122	939162

Filtering

Filtering Stage:

k-ton prime: prime p (or prime ideal $\wp$) appearing exactly in k relations with odd power

- ❶ delete 1-tons (singletons)
- ❷ merge 2-tons (doubletons)
- ❸ delete rows with more large primes
- ❹ delete heavy relations
- ❺ Above steps 1 – 4 introduce more 1-tons & 2-tons
- ❻ Repeat steps 1 – 4 until left with number of relations little more than total number of distinct primes that appear

Some Variations:

- *Adaptive Sieving*: Using larger factor base
- *Guided Sieving*: Mixing both Sieving & Filtering stages
- *Idea*: Avoid excess sieving

Adaptive Sieve

- Adaptive Factor base size:
Algebraic side: 15.34M prime ideals
Rational side: 07.0M primes

- Sieving regions: $A - B$ plane

	+ve B		-ve B	
	start	end	start	end
A	2^{29}	2^{30}	2^{29}	2^{30}
B	1	14720	1	2944

- Timings:
Sieving each block (16 B -lines, A -range $2^{29} - 2^{30}$) took ~ 8 hrs.
Total timings for line sieve: 1 CPU year.

Adaptive Sieve (Cont...)

- Yield:

Total number of Relations (rows) = 8207916

Total number of prime ideals = 15148091

Total number of primes = 7166136

Total number of columns = 22314227

- Data Analysis:

	Line Sieve	Line+Adaptive	Result
Relations	7793906	8207916	+ 5%
Singletons	13781089	13502535	- 2%

Guided Sieving

Given (a, b) , let $F_1(a, b)F_2(a, b) = \prod_p p^{\gamma_p}$

- **Rank of a relation (a, b) :**

$K_{k,l}(a, b) = \#$ of k -ton primes of size $\geq l$ bits in a relation (a, b) .

The rank $\mathcal{R}$ of relation (a, b) is defined by

$$\mathcal{R}(a, b) = \sum_{k=1}^{10} \frac{K_{k,20}(a, b)}{k}$$

Intuition: the deletion of a relation containing a k -ton results in introduction of new singleton with probability $1/k$

- **Weighted rank of a relation (a, b) :**

$K_{k,l}(a, b) = \#$ of k -ton primes of size l bits in a relation (a, b) .

. Then the weighted rank $\mathcal{W}$ of relation (a, b) is defined as

$$\mathcal{W}(a, b) = \sum_{l=20}^{30} w_l \sum_{k=1}^{10} \frac{K_{l,20}(a, b)}{l}$$

Intuition: Discarding relations with more large prime.

Guided Sieving (Cont...)

Effect of removal of relations in successive stages of 0.25 million.

# of relations	# Primes (rank $\mathcal{R}$) η_1	# Primes (rank $\mathcal{W}$) η_2	$\eta_1 - \eta_2$	# 1-tons left (rank $\mathcal{R}$) s_1	# 1-tons left (rank $\mathcal{W}$) s_2	$s_1 - s_2$
7100870	20747695	20747695	0	13176317	13176317	0
6850870	19737840	19731297	6543	12464676	12444588	20088
6600870	18848970	18836942	12028	11855506	11817486	38020
6350870	18013895	17997184	16711	11296573	11250105	46468
6100870	17201604	17196209	5395	10772303	10736484	35819
5850870	16447038	16429393	17645	10289059	10247676	41383
5600870	15707138	15680734	26404	9833843	9771204	62639
5350870	14971046	14946324	24722	9380099	9303560	76539
5100870	14253124	14221091	32033	8926923	8840806	86117
4850870	13549153	13511258	37895	8489432	8395915	93517
4600870	12852868	12807652	45216	8058921	7955347	103574

Table: Comparison of above two ranking of the relations

Lattice Sieve Method

- Much faster compared to line sieve.
- For a special prime q in the factor base FB compute the lattice s.t. q divides norm of each lattice point.

Important steps involved

- For given prime q compute the *shortest* basis vectors (u, v) . (Ext. GCD algorithm is used, involves large amount of multiplications and inverse computations.)
- Rest of the lattice points are the integer linear combination of basis vectors
- Efficient way needed to traverse through the lattice points

Lattice Sieve Method(Contd.)

- If the lattice basis satisfies certain conditions,² then all of the lattice points in the sieving area can be traced easily.
- Given the lattice basis (u, v) with 'certain conditions', and a current point u' , next point in 'certain vertical bound' will be one among $u' + u$ or $u' + v$ or $u' + u + v$.

For RSA-512

- Applied on singletons generated by Line+Adaptive sieve
- Time taken for 1 prime ~ 2 min
- 1000 primes yield 564 new relations
- For each block (16 B -lines) took ~ 3 hrs.

²J. Franke, T. Kleinjung, Continued fractions and lattice sieving; Proceedings SHARCS 2005;

Table 1 : Smoothness Probabilities: u^{-u} , $u = \log(nsz)/\log(psz)$
 norm size (nsz), prime base bound (psz) all in bits, number of primes (npr)
 table entries give the probabilities as negative exponents of 2
 (i.e. = expected number of trials to get one smooth number)

psz	26	28	30	32	34	36	38	40
npr	21.83	23.72	25.62	27.53	29.44	31.36	33.28	35.21
nsz								
128 :	11.32	10.02	8.93	8.00	7.20	6.51	5.90	5.37
160 :	16.13	14.37	12.88	11.61	10.52	9.56	8.73	8.00
224 :	26.77	24.00	21.66	19.65	17.92	16.41	15.09	13.92
256 :	32.49	29.19	26.39	24.00	21.93	20.12	18.54	17.14
320 :	44.57	40.17	36.43	33.22	30.44	28.02	25.89	24.00
384 :	57.37	51.81	47.08	43.02	39.50	36.43	33.72	31.33
448 :	70.77	64.00	58.25	53.30	49.02	45.27	41.96	39.04
480 :	77.66	70.28	64.00	58.60	53.92	49.83	46.22	43.02
512 :	84.67	76.67	69.86	64.00	58.92	54.47	50.55	47.08

Table 2 : NFS, QS Comparison: $L(N, \epsilon, c)$, $\epsilon = 1/3, 1/2$ all sizes in bits; estimates QSest, NSest: given as exponent of 2

deg	mvsz	tcsz	Arsz	Brsz	QSsz	NSrsz	NFasz	QSest	NSest
5	193	215	47	28	512	221	355	82.29	77.79
5	193	215	47	29	512	222	360	82.29	78.46
6	162	178	47	28	512	190	346	82.29	73.81
6	162	178	45	28	512	190	346	82.29	73.81
6	162	178	47	29	512	191	352	82.29	74.59
6	162	178	47	30	512	192	358	82.29	75.37
7	140	152	47	28	512	168	371	82.29	74.71
7	140	152	46	28	512	168	364	82.29	73.89
7	140	152	45	28	512	168	357	82.29	73.09
7	140	152	47	29	512	169	371	82.29	74.79
7	140	152	46	29	512	169	364	82.29	73.98
7	140	152	45	29	512	169	357	82.29	73.17
7	140	152	47	30	512	170	371	82.29	74.88
7	140	152	46	30	512	170	364	82.29	74.07
7	140	152	45	30	512	170	362	82.29	73.84

The Grand Challenge

What it Means and Takes

- A brief history of our earlier projects
- Walks, Runs and Jumps through NFS Arena
- Giant Killing - What it takes? : Estimates, Targets
- Tweaks and Treats- Analysis, New Science
- Bearing the Cross - Our Consortium
- Sharing the Burden - Distributed Math, Algo, Comptn
- Lending Hands and Shoulders - Synthesis and Splash

Integer Factoring (IFP), Discrete Logarithm (DLP) and Square Roots modulo Composites (SQRTP)

- The four stages of the IFP are: (i) polynomial selection for setting up a proper number field (ii) sieving - line, cubic, lattice and adaptive (iii) linear algebra - large, sparse, matrix system solver (iv) square roots of large algebraic integers.
- We have extensive experience with the mathematical, algorithmic, computational and practical aspects of all the four stages.
- We have been immersed deeply in this area during the last 10 years. We have completed all the four stages of the GNFS for factoring RSA moduli of sizes around 512 bits. This work was sponsored by MCIT. A related component work was sponsored by SAG.

- In this project we propose to address the finer computational issues of (i) the polynomial selection stage for large RSA moduli, between 512 and 1024 bits, (ii) special, adaptive schemes for sieving, (iii) efficient filtering schemes for the linear algebra stage, and (iv) fast computation of square roots of large algebraic integers.
- These will involve investigations into Newton identities, polynomial ring computations, polynomial roots over finite fields, high-speed computations with DFT and CRT, gcd and lattice basis computations, Lanczos, Wiedemann techniques, structured Gaussian elimination with graph theoretic models and algebraic number field computations.
- These investigations will include all aspects of design, analysis and large-scale implementation methodologies on large networks of cluster computing platforms.
- Finally, a consortium of groups will be mobilized to address the many sub-problems.

Enter The Integer RSA 1024

135066410865995223349603216278805969938881475605667027524485
143851526510604859533833940287150571909441798207282164471551
373680419703964191743046496589274256239341020864383202110372
958725762358509643110564073501508187510676594629205563685529
475213500852879416377328533906109750544334999811150056977236
890927563

A Trial Polynomial generated by my system

- $c_5=227155500$
- $c_4=3721524419596$
- $c_3=67896910213173842666472913784405940309785862188778736022$
- $c_2=383941979591092799583739726110749428811376518615606081724378$
- $c_1=287556506039325855692246065464116590265773025654302100906089$
- $c_0=101432980801639764217842930116716382435227095138218976431017$
- $m=901249051299261148574091032118886436371160147297009669866214$
- algebraic norm:
$$z = c_5 * a^5 + c_4 * a^4 * b + c_3 * a^3 * b^2 + c_2 * a^2 * b^3 - c_1 * a * b^4 + c_0 * b^5$$
- rational norm: $r = a + b * m$

Trial Polynomial - Quick Analysis

- $\log 2 = \log 2$, $\text{sizeof } m = \log m / \log 2 = 199.166$
- factor base $B_1 = q : q < 2^{40}$
- sieve size $A \simeq 2^{40}$, $B \simeq 2^{30}$
- rational norm size 170, algebraic norm size 360
- $ua = 360.0/40.0 = 9.0$, $ub = 170.0/40.0 = 4.25$
- smoothness probability:
 $\text{prob} \simeq ua^{-ua} * ub^{-ub} \simeq 10^{-11.26}$
- $\log(1/\text{prob})/\lg 2 \simeq 38.41$
- num primes 2^t , with $t = (40.0 - \log_2(40.0 * \log(2.0))) \simeq 35.21$
- num trials for full rank sieve matrix $= 2^{35.21} * 2^{38.41} = 2^{73.62}$
- with better polynomials, reduced norms sizes, larger factor bases, semi-smooth sieving with large-primes and lattice sieving, and various other tweaks, the expected sieving work would be 2^{75} cycles.
(i.e. 1 petaflop year!)

Strategies: Tricks and Tweaks

- 1. generate *good* NFS polynomials: neat skew, smooth coeff, rich roots, lean norms, high yield at low work, semi-numerical algo, (expected gain 2^2)
- 2. roots over finite fields- richness, adaptive line, lattice sieve aids (expected gain $2^{1.5}$)
- 3. reduce cycles per lattice point from 2^{10} to 2^6 (expected gain $2^{0.5}$)
- 4. line sieve with pos and neg values of b gives smaller norms (expected gain 2^1)
- 5. other tweaks in sieve organization- parallelism, distribution, general devices, special-purpose hardware (FPGA, ASIC) coordinated computing (expected gain 2^1)

Strategies: Tricks and Tweaks

- 6. Large prime variation - lattice sieve, adaptive sieve (expected gain 2^2)
- 7. Large primes: bi-composite, tri-composite factorizations - tuning $p-1$, $p+2$ and ECM (expected gain 2^1)
- 8. matrix filtering- row/col density models, cycles, graphs (expected gain 2^2)
- 9. linear algebra - theory, math, algo, implementation - Lanczos, Wiedemann, structured GE and combinations (?); serial/parallel (expected gain 2^2)
- 10. Square roots of algebraic integers (expected gain 2^0)

3.1. Traditional High Performance Computing

Top 500 - Nov 2025

Rank	Name	Site	Rating
1	El Capitan	LLN-USA	1.82 Ef
2	Frontier	ORN-USA	1.32 Ef
3	Aurora	ANL-USA	1.02 Ef
4	Julich	EU-Germany	1.00 Ef
5	Eagle	Microsoft	536 Pf
6	HPC6	EU-Italy	478 Pf
7	Fugaku	Japan	442 Pf
8	Alps	Switzerland	434 Pf
9	LUMI	EU-Finland	380 Pf
10	Tuolumne	US	241 Pf

3.1. Traditional High Performance Computing

- NSM, at Rs.4500 crores, during 2015 - 2022, is to deliver about 100 Pflops!!! (expected 16 Pflops, by Dec 2022).
- C-DAC Param-Sidhi, Feb 2021, is a 4.7 Pflops (my estimate Rs.200 Crores)
- IISc Param-Pravega, Feb 2022, is a 3.3 Pflops (my estimate of Rs.120 crores)
- C-DAC during 2017 - 2025, under the NSM, has set up in 37 institutions, an installed capacity of 40 Pf.
- C-DAC Bangalore, under the NSM, will set up a 30 Pf, supercomputing facility in 2026.
- C-DAC Bangalore, under the NQM, will set up a Rigetti-108, superconducting loop quantum computer in 2026.

3.1. Traditional High Performance Computing

A quick and rough guide to cost, speed and characteristics of ultra-large-scale computing resources

Supercomputers

0. Frontier 1000 Pflops = 1 exaflop, billion billion flops [2022]
(2^{85} flops year $\Rightarrow$ 1 RSA 1280 crack in 1 year!)
1. Fugaku 420 Pflops, IBM PanGea 25 Pflops are other machines [2022]
2. IBM, NVIDIA, US DoE announce in June 2018 the world's fastest supercomputer at ORNL, the Summit, of over 200 Pflops. Summit - 9216 Power 9- 22 core, 27648 Tesla V100 GPU, each with 6 NVIDIA Volta TensorCore GPU. \$ 200 M, (1Gflop/\$ or 15Mflop/Re - improving my guesstimate of 1 Mflop/Re), and a power rating of 15MW.

3.1. Traditional High Performance Computing

3. Sunway TaihuLight, by China National Research Center of Parallel Computer Engineering & Technology (NRCPC), with a High Performance Linpack (HPL) mark of 93 Pflops and a peak of 125 Pflops is the number one ranked machine. It has about 10.6 Mn cores with a power rating of 15 MW. The total cost of 1.8 Bn Yuan (US \$ 270 Mn) (Rs. 1700 crores) (improving my guesstimate to 6 Mflops for 1 Re).

4. The Chinese machine, Tianhe-2, had been the no.1 since 2013, at a thruput of 34 Pflops. It had been developed by 1300 scientists at Univ. SunYat-sen, Guangzhou, at Nat. Univ. Defense Technology. It has 3 Mn cores, power rating 18 MW, 1 PB memory, costing 2.4 Bn Yuan (400 Mn \$ US) (Rs. 2700 crores). (Note that my guesstimate of 1Mflop for 1 Re, is in sync with this figure of Rs. 27 Bn for 34 Pflops).

5. The next best, Titan of the US ORNL, at 18 Pflops, costs \$ 100 Mn (is about 0.25 of the above).

3.1. Traditional High Performance Computing

6. IISc machine SahasraT [2015] (CRAY XC40) comprises: (a) Xeon processor CPU cluster - a sustained throughput of 950 Tflops (b) GPU cluster NVIDIA K40 GPU cards - and 44 nodes sustained 52 Tflops (c) MIC cluster Intel Xeon-Phi 5120D - and 48 nodes sustained 28 Tflops. gives a TOTAL throughput of of 1 Petaflop, at a cost of Rs.80 crores
 $\Rightarrow 1.03 * 10^{15} / 80 * 10^7 \simeq 1.3 \text{ Mflops/Rupee.}$

7. IITroMet, NMRWFC machines, *Pratyush*, *Mihir*, jointly, (CRAY XC40 systems), (Jan 2018) deliver up to 7 Pflops. at a cost of Rs.450 crores
 $\Rightarrow 7.03 * 10^{15} / 450 * 10^7 \simeq 1.7 \text{ Mflops/Rupee.}$

8. NSM, at Rs.4500 crores, during 2015 - 2022, is to deliver about 100 Pflops!!! (expected 16 Pflops, by Dec 2022)

C-DAC Param-Sidhi, Feb 2021, is a 4.7 Pflops (my estimate Rs.200 Crores)

9. IISc Param-Pravega, Feb 2022, is a 3.3 Pflops (my estimate of Rs.120 crores)

4.1. Quantum Computing: Investment Trends

- *Inside Quantum Technology* (New York April 2020) : The future of quantum computing, quantum networking and quantum sensors - where quantum tech meets business
prediction 2020 market (US \$): QComm 140 M; QComp 110 M
- Jan. 2017 - Canada announced 4th stage \$ 0.5 B
- Jan. 2017 - Japan announced 3rd stage (from 4 yrs to 10 yrs) \$ 0.8 B
- Dec. 2017 - Australia (NSW) announced \$ 20 M (R 150 C) (5 yrs)
- Dec. 2018 - US announced \$ 1.2 B (R 9000 C) (5 yrs)
- Jun. 2018 - Europe announced \$ 1.13 B (R 9000 C) (5 yrs)
- Dec. 2018 - Russia announced \$ 0.8 B (R 5600 C) (5 yrs)
- Feb. 2020 - India announced \$ 1.12 (R 8000 C or R 80 B) (5 yrs)
- DST budget for 2020-21, R144 B, 10.9% increase over 2019-20
- China investments believed to be at least 10 times that of US

4.1. Quantum computing: Investment Trends

- observation in US : investment in, FFRD (Federally Funded Research and Development) is much lower *today* than BFRD (Business FRD), the total being about 3% of GDP (i.e about \$ 30 B).
(I deduce : $BFRD \text{ in US is } 2 * FFRD \implies \text{in QC, BFRD is } \$ 2.4 B$)
- observation in US : the life cycle cost of F35 is \$ 1.2 T, and the quantum investment is only 1 %.
- observation in US : China investment is at least 10 times, including R&D in Qcomm, QKD, stealth radars, SQUID (superconducting quantum interference detectors)
- observation in US : China is leading in QComm, QSensors, and US in QComp

4.1. Quantum Computing: Investment Trends

- Boston Consulting Group (8 reports during May 2018 - Mar 2021) estimates value \$ 450 Bn to \$ 800 Bn in the next 15 to 30 years and about \$10 Bn in the next 3 to 5 years.
- major applications and use cases are in
 - ① *simulation*: molecular design, drug discovery, battery design, derivatives and option pricing
 - ② *optimization*: route logistics, portfolio risk management
 - ③ *machine learning*: ML training with very large (millions) networks AI for autonomous vehicles
 - ④ *cryptography*: breaking traditional (classical) encryption and enabling stronger encryption, key management
- **Three stages of Development**
 - ① *NISQ*: noisy intermediate scale quantum computing - not error-corrected, useful for 1,2,3 above (next 5 to 10 years)
 - ② *Broad Quantum Advantage*: better error-correction, higher quality qubits, abstraction layers

4.2. Quantum computing: Science and Technology Trends

- *Roadmaps* (next 20 years): [Boston Consulting Group, Feb 2022]
 - Honeywell - **ion trap**¹ based, error-corrected, scaled model H1
 - IBM - pre-built, **superconducting qubits**³, quantum run-times integrated with circuit libraries
 - IonQ - **ion trap** based modular QCs networked in a data center
 - PsiQuantum - 1 million physical qubit system with a **single photon**³-based setup
 - Google - 1 million bit, error corrected, **superconducting qubits** system
- other near-future technologies:
 - quantum dots* (Intel, SQC) **electron/nuclear spin**⁴-based setup
 - cold atoms*: qubits are made from **neutral atoms**⁵, rather than ions, trapped by light and controlled by lasers

4.2.1. Algorithmic Complexity Strands

Classical Verification of Quantum Computations, [Urmila Mahadev],
[Quant-Ph, Sept. 2018, FOCS 2019 (Best Paper)]

- $BQP \subseteq QPIP_0$?
- first protocol for classical verification of efficient quantum computation.
- *measurement protocol* - classical verifier using a quantum prover as a trusted device
- prover: constructs n -qubit quantum state, reports measurements in a basis (e.g. Hadamard) of randomly chosen qubit by verifier.
- soundness based on LWE problem being quantum intractable
- computational assumption : LWE with super-poly noise ratio is BQP-hard
- completeness is based on certain distribution equivalence.
- uses : trap-door claw free families of functions; truncated discrete, Gaussian distributions over $\mathbf{Z}_q$
- LWE problem : distinguishing the distributions of a *ground random variate and its shifted version* in $\mathbf{Z}_q^n$ with samples from $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$.

4.2.1. Algorithmic Complexity Strands

Determining eigenstates and thermal states using quantum imaginary time evolution [Nature Physics, Nov 2019]

- quantum analogues of imaginary time evolution modeled by Schrödinger equation
- Lanczos, METTS algorithms for a new class of eigenstate, thermal state simulations
- can be carried out with no ancillae qubits or deep circuits, achieving exponential reduction in space, time
- may be useful for NISQ architectures to demonstrate quantum advantage

4.2.2. Physics and Engineering Strands

Topological Quantum Computation, [Freedman, Kitaev, Larsen, Wang], [Bull. AMS, V40, Oct.2002]

- anyons: quasi-particles (continuous phase states, between fermions and bosons) $\implies$ fault-tolerant qcomp
- knot theory, braid groups, topological quantum field theory, quantum Hall effect, quantum computation, Gallium Arsenide devices
- anyonic systems $\implies$ unitary topological modular functors, the basis of Jones polynomial (Knot theory), arise in Witten-Chern-Simons, quantum gauge field string, theory
- main advantage is physical error correction (over presumptive qubit models repairing combinatorially, requires much lower initial error)
- fault-tolerant TSC - [Aug 2019] new material "silicon" of QC - UTe₂

4.2.2. Physics and Engineering Strands

Photonic quantum info. processing, [Slusserenko, Pryde], [arXiv Jul 2019]

- PQC, LOQC
- decoherence-free quantum systems
- 5 out of 7 DiVincenzo criteria of a QC are satisfied; but (i) photons do not interact easily (deterministic 2 qubit gates hard) (ii) photon loss - generation, detection, scattering, absorption in optical components
- encoding by 2 modes of some degree of freedom (orthogonal polarizations, or overlapping propagation paths) (*dual-rail encoding*)

4.2.2. Physics and Engineering Strands

How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits, [Gidney, Ekerä], [arXiv Dec 2019]

- techniques from Shor [1994], Griffith-Niu [1996], Zalka [2006], Fowler [2012], Ekerä-Hastad [2017], Ekerä [2017, 2018], Gidney-Fowler [2019], Gidney [2019]
- planar grid of SC qubit platforms, physical gate error rate 10^{-3} , surface code cycle time $10 \mu\text{secs}$
- $3n + 0.002n \log_2 n$ logical qubits, $0.3n^3 + 0.0005n \log_2 n$ Toffoli gates, $500n^2 + n^2 \log_2 n$ measurement depth
- implications to RSA and DLP systems

4.2.2. Physics and Engineering Strands

Toward Scalable Quantum Computing Elements, IEEE Spectrum Jan. 2020

- cryogenic CMOS control
- cryogenic reconfigurable hardware
- cryo-CMOS device modeling
- micro low power relays - alternative to transistors
- rapid single flux logic (RSFL)
- Weyl semi-metal amplifier
- co-simulation of electronic control and quantum processor
- scalable fault-tolerant 2D, 3D qubit arrangements

4.2.2. Physics and Engineering Strands

Gallium phosphide as a piezoelectric platform for quantum optomechanics [Quant-Ph, Sep 2019]

- integrated piezoelectric optomechanical systems coupled to microwave circuits
- GaP crystal fabrication - 2.9GHz mechanical mode coupled to high quality factor optical resonator in the telecom band
- transduction of quantum information between different carriers

4.2.2. Physics and Engineering Strands

- **Encoding breakthrough allows for solving wider set of applications using neutral-atom quantum computers**, [Physics Today, 14 Feb 2023]
- **Quantum optimization with arbitrary connectivity using Rydberg atom arrays** [Quant-Ph, Sep 2022]
- *Optimization*: logistics, energy systems, biomedical, engineering design
- *Machine Learning*: automotive, pharmaceuticals, finance
- *Monte-Carlo simulation*: oil & gas, computational biology, chemistry
- *Big data analytics*: finance, healthcare, transportation
- *Material science*: biomedical, pharmaceutical, architecture

5. Post-Quantum-Public-Key-Cryptography- Algorithms [2022]

- quantum computation - primitives, algorithms [*cevmquantumlec1.pdf*]
- classical, quantum algorithms: IFP, DLP [*cevmquantumlec1.pdf*]
- Turing vs Quantum - *supremacy, advantage, co-existence, hybrid*
- post-quantum cryptography

Of the 26 finalists of the 2nd round NIST Selection Process, we consider the following 6 *representative* candidate submissions.

- | | |
|-----------------|---------------------------------|
| ① codes : | rqc (rank codes) |
| ② lattices : | pqNTRU (double rings) |
| ③ curves : | SIKE (super-singular isogenies) |
| ④ equations : | RAINBOW (MVQ) |
| ⑤ congruences : | NewHope (ring LWE) |
| ⑥ sequences : | SPHINCS (forest of hashes) |

5. Post-Quantum-Public-Key-Cryptography- Algorithms [2023]

- Four candidates, from Round 3, for Standardization

- ① Public-key encryption/KEM : CRYSTALS-KYBER (lattices)
- ② Digital Signatures : CRYSTALS-Dilithium (lattices)
- ③ Digital Signatures : FALCON (pqNTRU double ring lattices)
- ④ Digital Signatures : SPHINCS (forest of hashes)

- Candidates for Round 4

- ① Public key encryption/KEM : BIKE (Bit Flipping: QC MDPC codes)
- ② Public key encryption/KEM : HQC (Hamming Quasi Cyclic codes)
- ③ Public key encryption/KEM : Classic McEliece (structured codes)
- ④ Public key encryption/KEM : SIKE (super-singular isogenies)

CRYSTALS : Cryptographic Suite for Algebraic Lattices

(no discrete Gaussian sampling, most efficient polynomial convolution by NTT, small public key)

FALCON: Florida Atlantic University :

(efficient implementation)

5.1 Rank Quasi-Cyclic (RQC)

Structure

- encryption, key exchange
- $P \in F_q[X]$ of degree n . vector space, $\Psi : F_{q^m}^n \simeq F_{q^m}[X]/\langle P \rangle$
- vector product, associated ideal matrix, $(\mathcal{IM}(\mathbf{v}))$, generated by $\mathbf{v}, P$

$$\mathbf{u} \cdot \mathbf{v} = (u_0, \dots, u_{n-1}) (\mathbf{v}(X) \bmod P, \dots, X^{n-1} \mathbf{v} \bmod P)^T$$

- *rank weight* $\|\mathbf{x}\|$, of vector $\mathbf{x} \in F_{q^m}^n$ is the rank of the of the linear transformation matrix $M(\mathbf{x})$, whose entries are weights of the linear combination of basis elements of F_{q^m} , distance $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$.
- F_{q^m} linear code $\mathcal{C}$ of dimension k , length n is a subspace of dimension k of $F_{q^m}^n$ embedded with the rank metric, and denoted $[n, k]_{q^m}$
- *Ideal Code*: generator matrix made up of Ideal code matrices; closely related to *Quasi-Cyclic Codes*;
- unlike many linear codes under Hamming metric (such as RS), under rank metric only one family, *Gabidulin* codes have efficient decoding algorithms, using a q -polynomial reconstruction.

5.1 Rank Quasi-Cyclic (RQC)

Analysis

- operations *keygen*, *encrypt*, *decrypt* use random vectors private key and work in poly in q, n, m time algorithms.
- cryptographic strength is due to the hardness of *ideal rank syndrome decoding* and related problems, which require finding the closest codeword to a given vector (syndrome of received vector, or the received vector).
- classical hardness - syndrome decoding Gabidulin codes : (a) combinatorial techniques for finding support of error, or of codeword (b) algebraic attacks by Grobner basis techniques, all are exponential in q, n, m algorithms.
- quantum hardness - Grover search (square root of exponential)

5.2. pqNTRU

Structure

- $\mathcal{L} = Zb_1 + Zb_2 + \dots + Zb_d, b_i \in R^n$
- poly $f(x) = \sum_{i=0}^{n-1} f_i x^i$, $\|f\| = \sqrt{(\sum_{i=0}^{n-1} f_i^2)}$, $\|f\|_\infty = \max(|f_i|)$
- poly ring $\mathcal{R}_q = Z[x]/F(x)$, $F(x) = x^n \pm 1$.
- *modular lattice signature* : a lattice $\mathcal{L}$ with a trapdoor T , (a short basis of row vectors). The signature is $v = \sigma(m) = m \equiv \text{mod } p, v \in \mathcal{L}$.
- technical details are in rejection sampling, masking the trapdoor.

5.2. pqNTRU

Analysis

- some lattice hard problems are SVP (*shortest vector problem*), γ -SVP (approximate shortness), in terms of $\lambda_1(\mathcal{L})$, the first minima.
- generic hard problem is *short integer solution* in a lattice
- pqNTRU signature is an efficient instantiation of modular lattice signature over the NTRU lattice.
- another hard problem is based on learning with truncation (LWT) problem over the NTRU lattice.
- solving random LWT instance is as hard as solving certain worst-case instances of certain lattice problems.
- adapting Grover search to these gives square-root of exponential algorithm

5.2. pqNTRU: New algorithm for efficient NTRU keygen

- to solve for F, G , in $fG - gF = q \pmod{(x^n + 1)}$, for fixed f, g in the ring $\mathbf{Z}[x]/(x^n + 1)$. This is non-trivial.
- standard algorithms are at least quadratic in n for time, space
- New algorithm by [Pornin, Prest] [Feb 2022], uses a *field norm* in a *tower of fields* and achieve linear time complexity
- NTRU: given a monic polynomial $\phi \in \mathbf{Z}[x]$, of degree n , the lattice is generated by "short" polynomials f, g , with coefficients in $-1, 0, 1$. The polynomials f, g are *secret*, but their ratio $h = g/f \pmod{\phi} \pmod{q}$, for a given small integer (not necessarily prime) q is *public*.
- Schemes require the lattice to be *complete*, i.e. the private key owner requires two other short polynomials F, G , satisfying the NTRU equation $fG - gF = q$.
- NTRU lattices offer good performance.

5.2. Components of the new algorithm for pqNTRU

- repeated application of project-then-lift scheme (strategy in number theory and cryptanalysis)
- *towers of fields*:
 $K_I/K_{I-1}/\dots/K_0 = Q$, with $K_i = \mathbb{Q}[x]/(x^{2^i} + 1)$, and
- *associated towers of rings*:
 $\mathbb{Z}[x]/(x^n + 1) \supsetneq \mathbb{Z}[x]/(x^{n/2} + 1) \supsetneq \dots \mathbb{Z}[x]/(x^2 + 1) \supsetneq \mathbb{Z}$
- the field norm maps elements $f \in \mathbb{Z}[x]/(x^n + 1)$ onto a smaller ring. (this is exploited in attacks - map problem to smaller ring,, solve, lift)
- *New Idea* : the field norm works as a composition map in the tower :
 $N_{K/J} \circ N_{L/K} = N_{L/J}$
- use field norm to project equations over polynomial rings to equations over $\mathbb{Z}$ (descent phase), solve (faster over $\mathbb{Z}$), then lift solutions (lift phase)
- computations involve residue number systems, Chinese remainder theorem, cyclotomic fields, Galois conjugates, FFT, NTT - achieving linear time complexity

5.3. SIKE - Super-singular Isogeny Key Exchange

Structure

- $F_p, F_q = F_{p^2} = \{(a + i.b) : a, b \in F_p, i = \sqrt{-1}\}$
- Montgomery curve : $E_{a,b}(F_q) = \{(x, y) : by^2 = x^3 + ax^2 + x\} \cup O$, $b(a^2 - 4) \neq 0$, point addition, doubling, tripling formulae, abelian group, $\#E(F_q)$
- Hasse-Weil bound, m torsion, *super-singular* (i.e. $p \mid (q + 1 - \#E(F_q))$, j -invariant : $j(E_{a,b}) = 256((a^2 - 3)^3 / (a^2 - 4))$, unique up to isomorphism of the curve;
- SIKE protocol defines, a shared secret as a j -invariant of an EC.
- *isogeny* : $\phi : E_1 \rightarrow E_2$, is a non-constant rational map over F_q and a group homomorphism from $E_1(F_q)$ to $E_2(F_q)$.
- isogeny as two rational maps f, g , $\phi((x, y)) = (f(x), y.g(x))$
- subgroup isogenies : k -isogenies, $k = 2, 3, 4$; SIKE uses l^{e_l} , isogenies
- shared secret s is derived from the j -invariant of isogenous EC.

5.3. SIKE - Super-singular Isogeny Key Exchange

Analysis

- isogeny computation - arises in Schoof algorithm, in analysis of ECDLP; in quantum-safe crypto (although of much inferior performance than other algo)
- rich background use of ECC makes isogeny attractive alternative
- classical complexity [Galbraith et al] : $\tilde{O}(q^{1/4})$ for ordinary curves, $\tilde{O}(p^{1/2})$ for super-singular curves
- quantum complexity [Childs et al] : $L_q(1/2, \sqrt{3}/2)$. Here $L_q(\epsilon, c) = \exp(c \cdot (\log q)^\epsilon (\log \log q)^{1-\epsilon})$.
- for general curves, depends on : *class group of common Endo ring being abelian $\implies$ hidden abelian shift problem $\sim$ hidden subgroup problem $\implies$ applicability of Shor algorithm*
- for super-singular curves *class group of common Endo ring is not abelian*; best algo is based on quantum random walk, on l -isogeny graphs, and quantum search in $\tilde{O}(p^{1/2})$.
- recent quantum algo for ss curves [Biasse, Jao, Sankar]: $\tilde{O}(p^{1/4})$.

5.4. Rainbow - Multivariate Quadratic

Structure

- Private keys from three maps : two affine maps $\mathcal{S}, \mathcal{T}$, and a central map $\mathcal{F}$ from finite dimensional vector spaces over a finite field F_q . The central map is a set of multivariate quadratic polynomials.
- signature generation and verification are based on interleaved compositions of these maps.
- The m quadratic polynomials in the ring $F_q[x_1, \dots, x_n]$ have the structure of an unbalanced oil vinegar (UOV) scheme, with specified quadratic, affine variable indexing.
- The *random coefficients* are generated by cryptographic PRNG.

5.4. Rainbow - Multivariate Quadratic

Analysis

- classical hardness - hash collision, direct search, MinRank, HighRank, Rainbow Band Separation, UOV, all have the exponential bound q^v .
- resistance to 6 types of known attacks for signature forgery, key recovery.
- under-determined system ($n \equiv 1.5m$) of equations; Grobner basis; XL, Weidemann approach; (exponential in salient parameters)
- MinRank - linear combination of public polynomials of least rank.
- HighRank - linear representation of variables appearing least number of times in central polynomials (oil variables)
- quantum hardness - Grover search, equation solvers (square root of exponential in salient parameters)

5.5. NewHope - ring Learning With Errors

Structure

- rings, congruences, lattices
- $R_1 = \mathbb{Z}_q[x]/\langle f \rangle$
- random polynomial $a \in R_q$, secret, fixed, polynomial $s \in R_q$
- error polynomial e sampled from χ_σ
- $t = as + e \in R_q$
- search ring-LWE problem : given many (a_i, t_i) , find secret s
- encryption, decryption based on mapping, m to ring elements, further masking random elements, and inverting (with knowledge of secret) ring elements by decoding within error bounds on sampling

5.5. NewHope - ring Learning With Errors

Analysis

- ring learning with errors [Regev], under quantum reduction solving random LWE instance is as hard as solving certain worst-case instances of certain lattice problems.
- $\text{LWE} \simeq \text{learning parity with noise (LPN)} \simeq \text{hard decoding problem}$.
- solving a ring LWE instance means recovery of a secret vector $\mathbf{s} \in \mathbb{Z}_q^n$, given a sequence of approximate random equations on $\mathbf{s}$.
- straightforward algo: $2^{\Theta(n \log n)}$ (super-exponential) calls to SVP oracle. Certain type of lattice sieving reduces this to exponential.
- adapting Grover search to these, works in $2^{0.2n}$.

5.6. SPHINCS - hash based signatures

Structure

- hash based signatures - [Lamport, Merkle]
- SPHINCS - stateless hash based signature (earlier XMSS - stateful)
- few-times signature (FTS) key pairs, and many-times signature (MTS) key pairs organized on a hyper-tree (N -ary tree with d -levels)
- all key pairs are generated, in a Merkle-tree traversal scheme, from ONE key pair
- public keys are generated as end-elements of certain tweakable hash chains
- uses many indexing, operational, encoding, forest of random subsets data structures, hash families

5.6. SPHINCS - hash based signatures

Analysis

- conceptually easy, description of parameters, customizable options are complex
- analysis of collision probabilities, existential forgeries not complex
- exponential classical strength stems from the basic strength of the underlying hash functions and the binary liftings
- quantum strength is at worst reduced to square root of classical strength

6. Quantum Algorithms - IFP, DLP, ECDLP

Quantum Simulation: integer factoring, search

- polynomial time algorithms - quantum mechanics based model of computation- Shor for integer factoring and discrete logarithm and sub-linear algorithm Grover for unordered search
- sub-exponential, and linear algorithms for these problems in the conventional Turing machine model
- as of date, there is no effective quantum machine based implementation due to several practical obstacles.
- The quantum algorithms for factoring and discrete logarithm are based on the fundamental problem of finding square roots of unity, in $\mathbf{Z}_n^*$.
- *Quantum mechanics* is a mathematical framework for the construction of physical theories explaining many processes involving the fundamental atomic structures. A major advance in control of single quantum systems has given an impetus to the program of developing computation models for information processing.
- A *qubit* is a linear combination $\psi = \alpha|0\rangle + \beta|1\rangle$, of the two *computational basis states* $|0\rangle, |1\rangle$.

- Similarly, n qubits can be realized based on the 2^n basis states.
- Qubits can be realized by physical systems such as polarizations of a photon, alignment of nuclear spin in a magnetic field, states of an orbiting electron.
- Quantum computer is made of *quantum circuits* consisting of wires connecting *quantum gates*.
- A quantum gate is 2×2 unitary transformation of the qubit represented by the 2-dimensional vector (α, β) .
- There are many standard gates that operate on single and multiple qubits. These gates can be connected to form quantum logic circuits.
- *quantum parallelism* aids in the simultaneous evaluation of a function $f(x)$ for many values of x .
- Theory of computation is based on the postulates of quantum mechanics, the linear algebraic, function space (Hilbert space) and operator theory
- There is also a close parallel, to the computational complexity theory based on the Turing machine model applicable to the conventional algorithms, in the quantum world.

- A fundamental quantum algorithm is that of quantum Fourier transform which leads to phase estimation, order determination and integer factoring.
- The computations are described in the quantum circuit framework.
- Single qubit operations are characterized by unitary matrix transformations.
- Control logic such as *if A then B* are next expressed in terms of *controlled-NOT* (CNOT) gates.
- Multi-qubit operations are then introduced and represented by gating circuits and the corresponding matrix operators consisting of 2×2 unitary matrix blocks.
- Thus one builds a linear algebra of compositions of matrices representing compositions of gates.
- The CNOT and single qubit unitaries form a universal set for quantum computation. The set of unitary operations works over the continuous domain of complex numbers. We need to approximate the unitary operations by a discrete set of gates.

Quantum Computing Basics [Lipton, Regan, QC via LA, 2014]

- quantum algorithms rely are based on unitary transformation of unit vectors
- Two important families of transforms: Hadamard, Fourier
Hadamard: $H_N = H_{N/2} \otimes H$, $H_N[r, c] = \frac{1}{2^{N/2}}((-1)^{r \bullet c})$
Fourier : $F_N[i, j] = \frac{1}{\sqrt{N}}(\omega^{ij \bmod N})$
- transformations on vector a to get vector b :
Hadamard: $b = H_N a : b(x) = \frac{1}{\sqrt{N}} \sum_{t=0}^{N-1} (-1)^{x \bullet t} a(t)$
Fourier : $b = F_N a : b(x) = \frac{1}{\sqrt{N}} \sum_{t=0}^{N-1} (\omega)^{xt} a(t)$
- **quantum operations**: gates, permutation matrices, unitary operations
- **quantum operations**: flipping, copying, superposition, measurement

• quantum algorithms:

- ① *Deutsch* : Determine, in ONE evaluation, if a Boolean function $f : \{0,1\} \rightarrow \{0,1\}$ is a constant function.
- ② *Deutsch-Jozsa* : Determine, in ONE evaluation, if a Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$ is a constant function.

③ *Simon* :

Let $f : \{0,1\}^n \rightarrow \{0,1\}^n$ be a Boolean function. Let s be a hidden vector, such that $\forall y, z, f(y) = f(z) \Leftrightarrow y = z \oplus s$
if $s = 0$, then f is bijective, otherwise it is 2-to-1, f is periodic with period s . Simon's theorem and algorithm is that $\exists$ a quantum polynomial time algorithm to determine the period s .

④ *Shor* :

Let $f : \mathbf{N} \rightarrow \{0,1,\dots,M-1\}$, be a *feasibly*, computable function, with a *promise* that f is periodic with period r . i.e., $\forall f(x+r) = f(x)$. Shor's theorem and algorithm is that $\exists$ a quantum polynomial time algorithm to determine the period s .

⑤ *Grover* :

An element in a collection of N elements (unordered set), can be found in $O(N^{1/2})$ steps by a quantum algorithm.

- The *discrete Fourier transform* of a vector of complex numbers, $x_0, \dots, x_{N-1}$ is a vector of complex numbers $y_0, \dots, y_{N-1}$, defined by

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \exp^{2\pi ijk/N}$$

.

- The quantum Fourier transform, is similar. For an orthonormal basis $|0\rangle, \dots, |1\rangle$ of basis states, it is defined as an operator with the following action on the basis states

$$|j\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \exp^{2\pi ijk/N} |k\rangle$$

.

- Equivalently, the action on an arbitrary state is written as $\sum_{j=0}^{N-1} x_j |j\rangle \rightarrow \sum_{k=0}^{N-1} y_k |k\rangle$.
- The amplitudes y_k are the discrete Fourier transform of x_j .
- The Fourier transform is a unitary transformation and can be implemented as a quantum circuit.

- The Fourier transform is a principal element in the general procedure of *phase estimation* which is a main component of many quantum algorithms.
- The quantum Fourier transform has a useful *product representation*:

$$|j_1, \dots, j_n\rangle \rightarrow$$

$$\frac{(|0\rangle + \exp^{2\pi i 0.j_n} |1\rangle)(|0\rangle + \exp^{2\pi i 0.j_{n-1}j_n} |1\rangle) \dots (|0\rangle + \exp^{2\pi i 0.j_1j_2\dots j_{n-1}j_n} |1\rangle)}{2^{n/2}}$$

- Here the notation $0.j_1j_{l+1} \dots j_m$ represents the *binary fraction* $j_l/2 + j_{l+1}/4 + \dots + j_m/2^{m-l+1}$.
- An interesting aspect of the quantum Fourier transform algorithm is that the complexity reduces substantially, from the standard $O(n \log n)$ bound to $O((\log n)^2)$, an exponential gain in performance.
- The only catch, and a major one, is that the amplitudes in a quantum computer can not be directly accessed by measurement and also there is no easy way to *prepare* efficiently the original state which is to be transformed.

- The Fourier transform is central to the general procedure of *phase estimation* which is turn is important for many quantum algorithms.
- The phase estimation algorithm computes the phase ϕ of an eigenvalue $\exp^{2\pi i\phi}$ of the unitary operator U , given a corresponding eigenvector $|u\rangle$. This is by an inverse Fourier transform of the product state as

$$\frac{\sum_{j=0}^{2^t-1} \exp^{2\pi i\phi j} |j\rangle |u\rangle}{2^{t/2}} \rightarrow |\phi^*\rangle |u\rangle,$$

where ϕ^* is a good estimate for ϕ when measured.

- The integer factoring algorithm of Shor utilizes the phase estimation algorithm for the order determination problem, which gives a solution to the integer factoring problem.

Shor's Algorithm: Factorize $N = p.q$

- for random $a \in \{0, N-1\}$, find *period/order* r of a in Z_N^* , using $f_a(x) = a^x \bmod N$ (Note that $r \mid \phi(N-1)$)
- using rapid modular exponentiation prepare the functional superposition of $f_a(x) \forall x < N$
- run quantum part once and measure all qubits - the string formed by the first l , $l \simeq 2 \log_2 N$ yields an x in binary encoding (with high probability x is *good* (see definition))
- compute r from x by classical computation (continued fraction expansion of rational approximation)
- compute factorization using r, a , by $\gcd(a^{r/2} \pm 1, N)$

Good Numbers

For $Q = 2^l$, $N^2 < Q < 2N^2$, an integer $x \in \{0, \dots, N-1\}$ is **good** if $\exists t, \gcd(t, r) = 1$, such that $tQ - xr = k$, where $-r/2 \leq k \leq r/2$. (i.e., $tQ \equiv k \pmod{r}$) The lower bound on $\phi(r)$, is $\Omega(r/\log \log r)$. Choosing Q close to N , rather than N^2 , gives a high probability of finding a good x by choosing, randomly about $\log l$ values. This is not true for small r . However, the *Quantum Fourier Transform* drives the amplitude toward good numbers in all cases.

- Let $L(N) = \exp((\log N)^{1/3}(\log \log N)^{2/3})$. The NFS algorithm works in $L^{c+o(1)}$ with $c \simeq 1.93$.
- Recently Bernstein et al, propose low resource quantum factoring algorithm - fewer qubits, $(\log N)^{2/3}$ and time $L^{d+o(1)}$, $d = (8/3)^{1/3} \simeq 1.39$.
- This algorithm is asymptotically slower than Shor's algorithm, but requires much fewer qubits (about, 100 qubits for RSA1024 factoring!)
- The algorithm revisits the pre-quantum NFS algorithm and seeks to speed up the, crucial stage of *hunt for y-smooth* integers by a *quantum search* procedure.
- We look for a large number of *y-smooth* integers $x = (a + bm) * g(a, b)$
- The optimized parameters of NFS, lead to the search over integers x of size $(\log N)^{2/3+o(1)}$
- Prob (x is *y-smooth*) $\simeq u^{-u}$, where $u = \log x / \log y$. The estimation of smoothness probability has been considered by Ramanujan, Dickman, de Bruijn and others.

- The NFS sieving process, carries out this task in time linear in the search space of $L^{2\epsilon+o(1)}$.
- The main steps in this sieving stage are a combination of smoothness tests and explicit factorizations of *small integers* using ECM.
- Bernstein et al propose to replace these steps by the Grover quantum search algorithm to carry out smoothness tests and Shor's algorithm for factoring the smaller integers.
- The time is square root of the pre-quantum time, i.e. $L^{\epsilon+o(1)}$ and qubits reduce to $(\log N)^{2/3+o(1)}$.

- *low-resource quantum factoring*, [Bernstein, Biassé, Mosca, 2017]
- $O((\log N)^{2/3+o(1)})$ qubits, $O(L^{q+o(1)})$ time, where $L = \exp((\log N)^{1/3}(\log \log N)^{2/3})$, $q = (8/3)^{1/3} \simeq 1.39$
- *caveat 1* : *asymptotically*, with much *fewer* qubits, but much *slower* than Shor algorithm
- *caveat 2* : does not tell, whether RSA-2048 can be factored thus!
- *caveat 3* : the above theory is about *logical* qubits, *not physical* qubits (imposed by restrictions of error correction!)
- theory: [Gottesman - fault tolerant quantum computing], encoding Q logical qubits, T gates to fault-tolerant circuit using $Q(\log QT)^{O(1)}$, this bound is too loose, since $T = (\log N)^{1/3+o(1)}$, and the net estimate comes to $(\log N)^{O(1)}$
- *caveat 4*: to factor in $L^{q+o(1)}$, with sub-linear number of physical qubits, we need to encode Q logical qubits such that, for e.g.: using $Q^{1.49+o(1)}$ physical qubits, time overhead at most $\exp(Q^{0.49+o(1)})$, logical error rate at most $1/(\exp(Q^{0.51+o(1)}))$.
- the pre-quantum *multiplier* is $q > 1.9$

- *New algorithm features:*
 - ① fitting the product norm integer *requiring* only $Z = (\log N)^{2/3}$ or sub-linear bits, in Z qubits
 - ② NFS acceleration by quantum search:
find B -smooth $a + b * m$, the rational norm, and B -smooth $g(a, b)$, the algebraic norm, using Grover's search $\forall (a, b) \in \mathbf{ZXZ}$
 - ③ the NFS algorithm uses *roughly* $B = L$ bound as the smoothness bound or the factor base bound.
 - ④ the NFS *sieving* to find the smooth norms, requires *roughly*, the exploration of L^2 integers in $O(L^2)$ time
 - ⑤ the *BBM* algorithm uses Grover search in L time
- the algorithm seeks to find $F(s)$ between integers $-x, +x$, for l bit strings s , $\log y \in \Theta(l)$, $\log x \in (\log y)^2$, F is computable by a *reversible* $(\log x)^{1+o(1)}$ -bit quantum circuit in time $2^{o(l)}$
- it is possible to use the ECM factoring on these *smaller* integers, using Bennet quantum conversion strategy, and then Grover search
- however, the estimates are too conservative!!!
- the proposal is to use Shor algorithm for factoring the *smaller* integers
- the novelty is in using Shor's algorithm to a superposition of odd, positive integers, factoring them with significant probability

- Shor algorithm in superposition: (a) random selection of a , pre-computation $a^2 \bmod N$, (b) computation of the order r (function period, phase estimation of QFT), (c) continued fraction computation for the rational approximation of r , (d) final computation of $\gcd(N, a^{r/2} - 1)$.
- many technical, obstacles need to be handled - pre-computations of modular exponentiation powers, and storage within qubit budget; handling the analogs of large-prime situations (via $p - 1$ or ECM groups); parallel, serial constructions for recognizing smooth integers,

Short discrete logarithms and quantum Shor [Ekera, Hastad 2017]

- system state $|\Psi\rangle = \sum_{j=0}^{2^n-1} a_j \exp^{i\theta_j} |j\rangle$, $\sum_{j=0}^{2^n-1} a_j^2 = 1$.
- result of measurement of system collapses to a distinct state - probability of $|\psi\rangle$ to $|j\rangle$ is $|c_j|^2 = a_j^2$
- *quantum circuits* amplify the amplitudes of desired states, attenuate other states, thus enhancing the probability of collapse to a desired state
- *discrete quantum Fourier transform* (QFT) amplifies, by means of constructive interference
- QFT maps each state of a qubit register:

$$|j\rangle \longrightarrow_{QFT} \frac{1}{\sqrt{(2^n)}} \sum_{k=0}^{2^n-1} \exp^{(2\pi ijk/2^n)} |k\rangle$$

- Hence action of QFT on system is

$$|\Psi\rangle = \sum_{j=0}^{2^n-1} c_j |j\rangle \longrightarrow_{QFT} \frac{1}{\sqrt{(2^n)}} \sum_{j=0}^{2^n-1} \sum_{k=0}^{2^n-1} c_j \exp^{(2\pi ijk/2^n)} |k\rangle$$

- *constructive interference*: on observation the probability of collapse is $\frac{1}{2^n} \left| \sum_{j=0}^{2^n-1} c_j \exp(2\pi ijk/2^n) \right|^2$
- *claim 1* Let θ_j be phase angles, $|\theta_j| \leq \pi/4$. Then $\left| \sum_{j=0}^{N-1} \exp i\theta_j \right|^2 \geq \frac{N^2}{2}$
- DLP to find $d : x = g^d$, given g, x : (i) quantum stage - obtains "small number" s of good pairs (j,k) , (ii) classical post-processing stage - compute d
- *quantum algorithm* : Let m be least integer, s.t. $0 < d < 2^m$, let l be an integer close to m/s . The quantum algorithm returns a pair (j,k)
- with two registers of $l + m$ and l bits, let

$$|\Psi\rangle = \frac{1}{\sqrt{(2^{2l+m})}} \sum_{a=0}^{2^{l+m}-1} \sum_{b=0}^{2^l-1} |a\rangle |b\rangle |0\rangle$$

- compute $g^a \odot x^{-b}$ and store result in third register,

$$|\Psi\rangle = \frac{1}{\sqrt{(2^{2l+m})}} \sum_{a=0}^{2^{l+m}-1} \sum_{b=0}^{2^l-1} |a, b, g^a \odot x^{-b}\rangle$$

$$= \frac{1}{\sqrt{(2^{2l+m})}} \sum_{a=0}^{2^{l+m}-1} \sum_{b=0}^{2^l-1} |a, b, g^{a-bd}\rangle$$

- compute a QFT of size 2^{l+m} of the first register, a QFT of size 2^l of the second register to get

$$\Psi = \frac{1}{\sqrt{(2^{2l+m})}} \sum_{a=0}^{2^{l+m}-1} \sum_{b=0}^{2^l-1} |a, b, g^{a-bd}\rangle \xrightarrow{QFT}$$

$$\frac{1}{\sqrt{(2^{2l+m})}} \sum_{a,j=0}^{2^{l+m}-1} \sum_{b,k=0}^{2^l-1} \exp[2\pi i(aj + 2^m bk)/2^{l+m}] |j, k, g^{a-bd}\rangle$$

- observe the system in a measurement to get $(j, k), g^e$, where $e = a - bd$, with probability

$$\frac{1}{2^{2(2l+m)}} \left| \sum_a \sum_b \exp[2\pi i(aj + 2^m bk)/2^{l+m}] \right|^2$$

- the probability is large, or the success probability of e , the order estimate is good, for "good pairs", (j, k) , which by definition are pairs, where j , integer, such that $0 \leq j < 2^{l+m}$, and $|(dj + 2^m k) \bmod (2^{l+m})| \leq 2m - 2$
- a lower bound on the number of good pairs is obtained as follows:
 - $|\{j : \exists k, (j, k) \text{ is a good pair}\}| \geq 2^{l+m-1}$
 - probability of obtaining a specific good pair, by a single execution of the above algorithm, is at least 2^{-m-l-2}
 - Hence, probability of getting a single good pair, in a single execution is at least $1/8$

- compute d from s good pairs, $(j_i, k_i), i = 1, \dots, s$, using classical lattice based technique as follows:
- Let L be the integer lattice generated by the row span of the, $(s + 1)$ dimensional matrix of row vectors
 $\{(j_1, \dots, j_s, 1), (2^{l+m}, \dots, 0), \dots, (0, \dots, 2^{l+m}, 0)\}$
- Let $\mathbf{v} = (-2^m k_1 \bmod (2^{l+m}), \dots, -2^m k_s \bmod (2^{l+m}), 0) \in \mathbf{Z}^{s+1}$
- $\forall \mathbf{u} \in L, |\mathbf{u} - \mathbf{v}| < \sqrt{(s/4 + 1)} \cdot 2^m$ test if the last component of $\mathbf{u}$ is d , by computing if $x = g^d$.
- since s is constant, close vectors to $\mathbf{v}$ can be found *efficiently*
- all the above results are put together, to establish the Theorem:
 Let d be an integer, $0 < d < 2^m$, $s \geq 1$, fixed integer, l an integer close to m/s , g a generator of a finite, cyclic group of order $r > (2^{l+m} + 2^l d)$. Then there exists a quantum algorithm that yields a pair as output, when executed with $x = g^d$ as input. The main operation is an exponentiation of g in superposition to an exponent of length $l + m$ bits. If this quantum algorithm is executed $O(s)$ times to yield a set of pairs S , then there exists a polynomial time classical algorithm that computes a value d , when executed with all unordered subsets of $s \in S$.

- several optimization steps, parameter choices, quantum computer architectures matter in the efficient implementation
- applications are in Diffie-Hellman key exchange, short exponent based exponentiation schemes (practical considerations motivated by balancing the sub-exponential, computational complexity of generic NFS, versus, the square root complexity of the generic Pollard-rho, Pohlig-Hellman lifting methods)
- application to factoring RSA modulus is as follows: Let $N = pq$, G a cyclic sub-group of $\mathbf{Z}_N^*$, then G has order $\phi(N)/t$ for some $t|\phi(N)$, such that $t \geq \gcd(p-1, q-1)$. Assume, w.l.o.g, that $\phi(N)/t > (p+q-2)/2$. Let g be a generator of G .
- compute $x = g^{(N-1)/2}$. Then $x \equiv g^{p+q-2}$.
- compute the short discrete logarithm $d = (p+q-2)/2$, from g, x .
- compute p, q by solving the quadratic equation $N = (2d - q + 2) * q = 2(d+1)q - q^2$.
- Note that we obtain a short logarithm, because, $N - 1 = pq - 1 = (p-1)(q-1) + (p-1) + (q-1)$, and if $\phi(N)/t > (p+q-2)$, then $(N-1)/2 \equiv (p+q-2)/2 \pmod{\phi(N)/t}$.

- clearly, g can be chosen, uniformly randomly in the interval $1, \dots, N-1$
- to compute the short discrete logarithm, the above algorithm is used effectively, meeting the requirement on the order:
 $\phi(N)/t > 2^{l+m} + 2^l d \Rightarrow \phi(N)/t \geq 2^{l+m+1}$, since $0 < d < 2^m$.
- Note that $2^n \leq d = (p+q-2)/2 < 2^{n+1} \Rightarrow m = n+1$
- Further, note that $\phi(N) = (p-1)(q-1) \geq 2^{2(n-1)}$ which implies $\phi(N)/t \geq 2^{2(n-1)}/t \geq 2^{l+m+1} = 2^{l+n+2} \Rightarrow t < 2^{2(n-1)-(n+l+2)} = 2^{n-l-4}$.
- Since, $l = m/s = (n+1)/s, s \geq 1$, for random p, q , and a randomly selected cyclic subgroup of $\mathbf{Z}_N^*$, the requirement $t < 2^{n-l-4}$ is met with high probability for any $s \geq 1$.

Quantum ECDLP Resource Estimate: [Roettler,Naehrig,Swore,Lauter, Sept 2017]

LIQ*U*i|⟩: **Software design architecture and domain-specific language for quantum computing** [Wecker, Svore, Feb.2014] is a *Language Integrated Software Operations* system, with in-built, robust language **F#**

- n -bit prime field, $9n + 2 \log_2 n + 10$ qubits, quantum circuit of at most $448n^3 \log_2 n + 4090n^3$ Toffoli gates.
- *core* : Toffoli networks for controlled EC point addition, for Shor algorithm: Toffoli gate $|x, y, z\rangle \mapsto |x, y, z \oplus xy\rangle$ is a universal, reversible gate.
- gate-level comparison of Shor factoring: $\implies$ qubits for current classical ECDLP security level is *smaller* than for RSA.
- quantum models: QTM, quantum circuit model, quantum adiabatic model, QRAM model.
- LIQ*U*i|⟩ requires a hybrid architecture - quantum hardware and classical computer, modeled on quantum circuit and QRAM models, with interfaces to adiabatic evolution on quantized Hamiltonians.

Factoring with $2n + 2$ qubits and Toffoli based modular multiplication [Haner, Roettler, Svore, QIC, 2017]

- reversible algorithms for modular quantum arithmetic on quantum registers.
- modular multiplication - (a) modular doubling, addition (b) Montgomery multiplication (i.e., residues modulo p are mapped to $R^j p$, for radix R and power j , for fast multiplication by shifting).
- cost of additional *ancillae* bits (auxiliary working bits) is traded for speed.
- also modular inverse, is speeded using Montgomery inverse, via binary extended Euclidean gcd algorithm.
- all reversible circuits implemented in $F_{\#}$ using LIQUI—)
- a central operation in Shor algorithm: modular exponentiation of a constant a , by a superposition of values of x , stored in a register of $2n$, $n = \lceil \log_2 N \rceil$ qubits, with an output register initialized to 0, as given by, $|x\rangle|0\rangle \mapsto |x\rangle|a^x \pmod{N}\rangle$.
- the similar operation in the case of ECDLP is the scalar multiplication.

Resource Estimates for Shor algorithm for ECDLP versus Factoring

[Roettler,Naehrig,Swore,Lauter, Sept 2017]

$\lceil p \rceil$ bits	qubits	Toffoli gates	Toffoli depth	time sec	$\lceil \log_2 N \rceil$ bits	qubits	Toffoli gates
110	1014	$9 \cdot 10^9$	$8.66 \cdot 10^9$	273	512	1026	$6 \cdot 10^{10}$
160	1466	$3 \cdot 10^{10}$	$2.73 \cdot 10^9$	711	1024	2050	$6 \cdot 10^{11}$
256	2330	$1 \cdot 10^{11}$	$1.16 \cdot 10^{11}$	3848	3072	6146	$2 \cdot 10^{13}$
384	3484	$5 \cdot 10^{11}$	$4.05 \cdot 10^{11}$	17003	7680	15362	$3 \cdot 10^{14}$
521	4719	$1 \cdot 10^{12}$	$1.05 \cdot 10^{12}$	42888	15360	30722	$3 \cdot 10^{15}$

Period finding with adiabatic quantum computer [Itay Hen, May 2014]

- Adiabatic Quantum Computer (ADC) : the device is a *quantum annealer* (e.g. D-Wave machine), well suited for optimization problems.
- solution to an optimization problem is encoded in the ground state of a 'problem' Hamiltonian $\hat{H}_p$; system is prepared in the ground state of a 'drive' Hamiltonian $\hat{H}_d$
- in the Quantum Adiabatic Algorithm (QAA), the Hamiltonian of the system is slowly modified from $\hat{H}_d$ to $\hat{H}_p$ using linear interpolation: $\hat{H}_s = s\hat{H}_p + (1-s)\hat{H}_d$, with a smoothly varying $s(t)$, $0 < s(t) < 1$.
- a slow evolution, (in a magnetic field), keeps system close to the ground state of $\hat{H}_p$ and a final measurement gives the solution with high probability.
- $s(t)$ is the *adiabatic* annealing parameter.
- AQC and circuit based computations are polynomially equivalent [theory].
- However, only Grover search, quantum counting problems have led to successful QAA.

- a central problem is the *hidden subgroup problem* (HSG): determining the generating set of the subgroup H of a finitely generated group G , given that the function mapping the group G to a finite set S is constant on cosets of the subgroup F .
- the simplest HSG is the Deutsch-Josza problem;
- this paper develops a QAA for the more general Simon problem, which is central to Shor algorithms.
- in Simon's problem we seek to find the period of a function from n -bit strings to $(n - 1)$ -bit strings, that is constant only over inputs that are related by an unknown XOR mask (the period).
- a simpler, HSG problem, is the Bernstein-Vazirani (BV) problem: given an oracle that evaluates the function:

$$f(w) = \sum_{k=0}^{n-1} w_k a_k \pmod{2}$$
 where, w_k, a_k are the bits of two integers w, a , it is required to find the bits of a , with few queries to the oracle.
- establishes the possibility of AQC for Shor algorithm - much work is needed on (a) a 'real' periodic function, (b) adiabatic algorithm for QFT, and (c) a many-qubit quantum annealing system.

Optimizing Shor Simulation

- A.Dang, C.D.Hill and L.C.L Hollenberg, Optimizing matrix product state simulations of Shor's algorithm, (arXiv:1712.07311v3[quant-ph] 2 July 2018).
- mapping the entangling properties of the circuit over the one-dimensional structure of a matrix product state.
- space requirements normally depend on *order* r . In this work it depends on the factors of r .
- state requires 2^n complex scalars; by matrix product representation of quantum state can reduce this to the *amount of entanglement*.
- further optimization: w.r.t. single qubit measurement to collapse the entanglement.
- nontrivial instance of Shor algorithm on a supercomputer: 60 qubits, 14 TB RAM (against straightforward $2^{60} * 128\text{bits} \simeq 16 \text{ ExaB}$. (about 10^7 reduction!!!)

Matrix Product State (MPS) Representation of Pure Quantum States

- D.Perez-Garcia et al, Matrix Product Representations, (arXiv:quant-ph/0608197v2 14 May 2017).
- classical simulation of circuit model: set of 1, 2 qubit gates, applied to a number of qubits, in initial pure set and measured separately at end.
- in the cluster model: a multipartite state is prepared initially and with single qubit measurements at end.
- for both models, MPS formalism provides an understanding of the need for large entanglements for exponential speedup.
- Theorem 19. (Simulating quantum circuits): If at every stage in a polynomial time circuit quantum computation has a MPS representation with (bond dimension) $D = O(\text{poly}(N))$, then the computation can be simulated classically with $\text{poly}(N)$ resources. This is particularly true if every qubit-line in the circuit is crossed or involved by at most $O(\log N)$ two-qubit gates.

7.0 Lattice Geometry based Classical Factoring

- *Fast Factoring Integers by SVP Algorithms, corrected* by C.P.Schnorr, [work in progress, 16 Jun 2021]
- N , n -th prime, p_n , p_n smooth triples $\langle u, v, |u - v * N| \rangle$ for small, random, multipliers u, v
- obtained from *approximate short vectors* in lattice denoted $\mathcal{L}(\mathbf{R}_{n,f})$, where dimension of lattice is n , and f a permutation $f : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$. Let $N' = N^{1/n+1}$.
- The lattice matrix consists of $f(1), \dots, f(n), N' \log N$ in the diagonal and $N' \log p_1, N' \log p_2, \dots, N' \log p_n, N' \log N$ in the last line.
- Then the estimated work factor is
for n , $N \equiv 2^{400}$, $n = 47$, $ops = 4.2 * 10^9 \approx 2^{33}$,
for n , $N \equiv 2^{800}$, $n = 97$, $ops = 8.4 * 10^{10} \approx 2^{37}$

7.0 Lattice Geometry based Classical Factoring

- *integer linear combinations of vectors over reals, basis matrix, dimension, volume, Minkowski distance*
- *bounds on length of shortest vector*
- *Generic normal form, Gram-Schmidt ortho-normalization,*
- *QR decomposition, HKZ basis (Hermite, Korkine, Zolotareff)*
- *reduced basis, LLL algorithm*
- *SVP, CVP, approximate SVP, CVP are NP-hard*
- *relative density, efficiency of algorithms*
- *estimates of Schnorr on lengths approximate short vectors, lattice volume bounds determine the smoothness bounds !!!*
- *these estimates are being contested*
- *hence scaled quantum algorithms, based on lattice geometry, as efficient alternatives to Shor algorithm is not clear yet !!!*

7.1 Lattice geometry based Quantum Factoring

- NISQ; low resource qubits, circuit depth Quantum advantage??
- variational QA (classical + quantum) hold promise during NISQ :
example quantum approximate optimization algorithm (QAOA) for
eigenvalue, ML, molecular simulation
- Shor's algorithm on a fault tolerant QC is resource intensive
largest integer factored using Shor is 21 [2001, 2012, 2016]
- Estimate: 2048 bits, 8 hrs, 20 M noisy qb [Gidney, Ekerä 2021]
Estimate: 2048 bits, 177 days, 13436 noisy qb [Gouzien et al 2021]
- *Factoring integers with sub-linear resources on a superconducting
processor*, [arXiv:2212.1237.v1][quant-ph] [23 Dec 2022],
a 24 author paper from China.
- universal quantum algorithm for IFP
- combines classical lattice reduction with a quantum approximate
optimization algorithm (QAOA)

7.1 Lattice geometry based Quantum Factoring

- classical Schnorr factoring algorithm by a QAOA
- preprocessing to convert to closest vector in a lattice (CVP)
- then a quantum computer solves the optimization problem to refine the classical vectors computed by Babai algorithm;
- optimized results are then fed back to Schnorr algorithm
- post processing outputs the primes p, q
- Result : factored:
1961(11b), 3 qb; 48567227(26b) 5 qb, 261980999226229(48b) 10 qb
- uses Babai algorithm optimizing the size-reduce step in lattice basis reduction through QAOA;
- $O(\log N / \log \log N)$ qubits - sub-linear in length of N
- demonstrate factoring integers up to 48 bits with 10 SC qubits
- estimate 372 physical qubits and depth of thousands for 2048-bit N
- shows promise for realistic factoring
- this can be used for lattice reduction problems in PQC candidates !!!

7.1 Lattice geometry based Quantum Factoring

classical + quantum hybrid framework

- ① **classical**: integer N to random CVPs on an n dimensional lattice
- ② **classical**: [Schnorr]: L^3 / Babai algorithm $\rightarrow$ approximate orthogonalization $\rightarrow$ approximate solutions to CVPs
- ③ **classical**: [Schnorr] mapping of CVP lattice vectors to smoothness relations $u_i, v_i, |u_i - v_i N|, 1, \dots, n$ are p_n smooth,
- ④ **quantum**: Hamiltonian optimization : L^3 / Babai solution $\rightarrow$ QAOA solution (Adiabatic / circuit quantum processor), annealing, rounding
- ⑤ **classical**: identification of subset of approximate lattice vectors with smooth relations
- ⑥ **classical**: GF_2 linear algebra solver (Gaussian elimination / Lanczos / Weidemann) $\rightarrow$ solutions to quadratic congruences
- ⑦ **classical**: final gcd computation to determine $(p, q) : N = p * q$

8.0 Super-singular Isogeny based DH, KE of PQC

Preliminaries

- elliptic function from geometry, elliptic integral from analysis,
- related many power-series, analytic functions, modular functions,
- elliptic curve over Z, Q, R, F_q , rational points, Diophantine equations, modular forms, modular series including Ramanujan functions
- EC $E(F_q)$ over a finite field F_q ; cubic curve $y^2 = x^3 + a * x + b, \mathcal{O}$; addition law; genus;
- algebraic equations: Weierstrass form, affine form:
- order of group $E(F_q)$, Hasse-Weil theorem, order of elements
- j -invariant; Δ, η, τ functions
- isomorphism, homomorphism, endomorphism, isogeny
- k -regular graphs; eigenvalues; spectra; expander graphs
- *super-singular isogeny graphs or Ramanujan graphs*

8.0 Super-singular Isogeny based DH, KE of PQC

Examples

- projective coordinates:

$$E : Y^2 + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3$$

- affine form : $X/Z = x$; $Y/Z = y$:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

- Let $p = 2003$, $a_1 = 2a_3 = 8$, $a_2 = 5$, $a_4' = 1136$, $a_6 = 531$,
 $\rightarrow \Delta = 1707 \neq 0, j = 171$
- $P_1 = (1118, 1493)$, $Q_1 = (892, 591) \in E(F_p)$; $P_1 \oplus Q_1 = (1681, 1706)$, $2P_1 = (1465, 677)$
- By fast scalar multiplication algorithm we get $763P_1 = (143, 1428)$
- $|E(F_p)| = 1956 = 12 * 163$, the point $P_1 = (1118, 269)$ is of order 1956, which implies the group is cyclic generated by P_1 . The point $Q_1 = (892, 529)$ is of prime order 163.
- $E(K)$, kernel $E[n] = \{P \in E(K) \mid nP = P_\infty\}$, any point $P \in E[n]$ is called a n -torsion point.

8.0 M-SIDH, MD-SIDH : countering SIDH attacks by Masking

- SIDH- super-singular isogeny Diffie Hellman design [Jao, De Feo, 2011]
- underlying SIKE to NIST 4th round in May2022
- meet-in-the middle attacks of exponential complexity
- picture dramatically changed in Aug 2022 - Castryck-Decru attack and Maino-Martindale and Robert - exploit torsion point information in the protocol
- TWO counter measures [Fouotsa, Moriya, Petit Sep 2022] - masking isogeny degree information (MD-SIDH) and masking torsion point information (M-SIDH)
item parameters M-SIDH : PK: 4333, 7037, 9750 bytes for NIST security levels 1, 3, 5.
- isogenies : maps between $E, E' : \text{EC over } F_q$; isogenous curves; ϕ from E to itself is called *endomorphism* of E ; the set of all endomorphism with the 0 map is called *endomorphism ring*
- TWO types of curves:
(a) for ordinary curves : It is an order in a imaginary quadratic field (commutative)
(b) for super-singular curves: It is a maximal order in quaternion algebra (non-commutative)

- Diffie-Hellman protocol straightforward in ordinary curves:
starting curve E_0 ; Alice $\phi_A : E_0 \rightarrow E_A$; Bob $\phi_B : E_0 \rightarrow E_B$; A and B exchange E_A, E_B , computes the secret isogeny from the other's public key; since the isogenies are commutative they get the same curve E_{AB} (up to isomorphism); they use the j -invariant of E_{AB} as their shared secret.
- the isogeny $E_A \rightarrow E_{AB}$ computed by A is not the same as ϕ_A , as their domain and co-domain are different; hence denoted ϕ'_A
- But we have $\phi'_A \circ \phi_B = \phi'_B \circ \phi_A$
- with super-singular curves DH protocol is more difficult since the endomorphism rings are non-commutative
- A novel idea for SS curves by [Jao, Feo 2011] follows:
 - Fix co-prime degrees A, B of isogenies ϕ_A, ϕ_B
 - Alice reveals the images of a basis of the B -torsion of E_0 (i.e $E_0[B]$) and Bob reveals the images of a basis of the A -torsion of E_0
 - leads to SIKE, compact key sizes; most protocols use a variant of the isogeny problem : attacker is provided with additional non-trivial information: the degree of isogeny, and the images of torsion points. Then the problem is

- Problem 1: (Pure Isogeny) : Given 2 isogenous EC E, E' , compute an isogeny from E to E' .
- Problem 2: Let E_0 be a SS curve over $F_{p^2}, p = AB - 1$. Set $E_0[A] = \langle P, Q \rangle$. $\phi : E_0 \rightarrow E'$, be an isogeny of degree B . let $P' = \phi(P), Q' = \phi(Q)$.
Given E_0, P, Q, E', P', Q' compute the isogeny ϕ .
- in SIDH, SIKE : starting curve E_0 is special $\rightarrow$ its endomorphism ring is known
- this was exploited, by [Petit, 2017], assuming $A \ll B$ to set up a poly time attack
- Much faster, practical attacks were devised by [Castricky, Decru Jul 2022], followed by [Maino, Martindale], and [Roberts] for arbitrary starting curves
- these all exploit both torsion point information and the degree of the secret isogeny
- *saving grace* : $\exists$ many secure isogeny protocols in which no torsion point information is revealed [CRS, CSIDH, CGL, GPS ...]
- in Sep 2022, [Fouotsa, Moriya, Petit] announced countermeasures: by masking information
- MD-SIDH :idea1: degrees A, B of the secret isogenies are not fixed: they are uniformly random divisors of A, B

- MD-SIDH idea2: to defy pairing based recovery and DL computations over smooth order groups, the images of the torsion points are scaled by random integers
- M-SIDH only idea 2 is used and another idea3: isogeny degrees A, B , will have $t \geq 2\lambda$ distinct prime divisors. This makes recovering the scalar multiplier difficult, even though its square modulo A , or B is known.

Many techniques used in the security, performance analysis require:
computation and construction of isogenies
recovering isogeny degree up to squares;
square roots modulo N composite;
number of such problems given by distinct prime divisors of N ;
their solutions modulo primes;
smooth neighbours of primes
Chinese remainder theorem based lifting

⋮

*If computers that you build are quantum,
Then spies everywhere will all want 'em.
Our codes will all fail,
And they will read our email,
Till we get crypto that's quantum, and daunt 'em.*

Jennifer and Peter Shor 1994

*To read our email, how mean
of the spies and their quantum machine;
be comforted though,
they do not yet know
how to factorize twelve or fifteen.*

V.Strassen 1995

Can Committees Cultivate Code Crackers?

*The answer is an imponderable **qubit** at large.
Sums of powers of moderate members do converge,
although not all views align in the same area,
unlike the conjectured conduct code of zeros of zeta.
A standing committee is a sitting duck on an average.*

cevm 2016

Power of Quantum

*It isn't easy to make practical crypto crackers.
by all the King's mathematical, statistical hackers,
and all the Queen's computational, physicist nerds
until they get together to build SCQ machines in herds,
keeping entanglements, sieves, superpositions, free of errors.*

cevm 2022

Quo Vadis Quantum Computing?

*From qubits to superpositions and entanglements,
via trapped ions to cold atoms and photon alignments-
Key labs of the world race in the quirky quantum forest,
to search faster, predict better and optimize best.*

Will crypto cracking be worthy of quantum denouements?

cevm Apr 2024

Quo Vadis Quantum Factoring?

*I factored RSA640, in 2008, with NFS and cluster compute of
120 Gflop-years;
my, unfulfilled, wish was RSA1024 crack by 2024 with MNFS
in 4 Pflop-years.
While super-conducting loops, quirky spins, trapped ions, pho-
ton alignments,
are all in a global race to lead to useful quantum computing
grand denouements,
I predict, an RSA2048 crack before 2048 is full of quantum
noisy nightmares!*

cevm 12 Dec 2024 : $12+12+2024=2048$

Quo Vadis Q—nt-m E—or Co—e—ion?

*Logical to physical qubits ratios, and circuit modularity,
are entangled obstacles to quantum advantage and utility.
Classical to quantum error detection, correction jumps,
are doubly challenging due to magnitude and phase flips.
Yet, squeezing cats, concatenation, may be good checks on error
parity.*

cevm 14 Apr 2025 : World Quantum Day

Quo Vadis CQHPCCC?

*Machines and models from Abacus to Babbage and Turing to Feynman,
have raised compute performance many fold, hopes have risen
among both savvy friends and adversaries
and their cryptanalysts and cryptographers
to continue forever to outdo each other as a routine.*

cevm 02 Nov 2025

Evanescesne ex quantum, RSA 2048, prius 2048?

Will you fade away due to quantum, RSA2048, before 2048?

I hope not, but we think yes.

*As all super-positioned qubits, and coherence
by NISQ annealing or large gating circuits
need to reach high quantum error tolerance.*

*I factored RSA640, in 2008, using our tweaked Nfsieve
on a cluster+HPC setup in about 120 Gflop-years;
We estimated RSA1024 crack in machine cycles 2^{75} ,
a classical compute goal, in view of records of 768, 840 bits.*

*Super-conducting loops, and quirky spins,
are on a race to build a useful quantum computer,
along with photon alignments, and trapped ions.
Will you, RSA2048, fade quantumly, by 2048 or before?*

*Now from Gidney $\pm$ Regev $\pm$ Ragavan and Vaikuntanathan,
 $\pm$ Shor et al, I see, a hybrid is the only route to destination.*

Shakespearean Sonnet Form:

pattern: 3 quatrains, 1 couplet

rhyme : a-b-a-b c-d-c-d e-f-e-f g-g

meter : iambic penta/hexa meter

$$\text{cevm } 21+01+2026=2048$$