Post Quantum Cryptosystems

Abrahim Ladha

November 12, 2019

Stuff

- Background
- Other schemes
- Hash and block cipher based
- Isogeny/Supersingular based
- Lattice Based
- Code Based

Scheme	Lattice	Code	Isogeny	Rank
BIKE		✓		
Classic McEliece		✓		
CRYSTALS-Kyber	✓			
FrodoKEM	✓			
HQC		✓		
LAC	✓			
LEDAcrypt		✓		
NewHope	✓			
NTRU	✓			
NTRU Prime	✓			
NTS-KEM		✓		
ROLLO				✓
Round5	✓			
RQC				✓
SABER	✓			
SIKE			✓	
Three Bears	✓			

Scheme	Lattice	Multivariate	Other
CRYSTALS-Dilithium	✓		
Falcon	✓		
GeMSS		✓	
LUOV		✓	
MQDSS		✓	
Picnic			✓
qTESLA	✓		
Rainbow		✓	
SPHINCS+			✓

Evaluation Criteria

- Security, Cost/Performance, Algorithm characteristics in that order
- Security: IND-CCA2 (signatures EUF-CMA)
- Cost: space, time, probability of decryption error
- Characteristics: Systems it can run on, parallizability, elegance

Paramter Levels (at least as hard as)

- Level1: AES128 exhaustive key search
- Level2: SHA256 collision search
- Level3: AES192 exhaustive key search
- Level4: SHA384 collision search
- Level5: AES256 exhaustive key search

Mailing List interesting attacks

- Compact LWE: Plaintext recovery only from ciphertext. Modified by authors but same vulnerability
- CFPKM: Plaintext recovery with ciphertext and public key
- DRS Signature scheme: uses a loop. Most inputs will halt in $< 10{,}000$ iterations, but $1/30$ will loop forever.
- DME: Lots of complicated algebraic attacks
- Giophantus: IND-CPA with $1/(10\sqrt{n})$ faster than decryption
- "Random polynomial"
- Some removed with no clear comments as to why (maybe NIST internal cryptanalyst)

A lot of exotic systems

- Merkle-Hellman [MH78] (Knapsack)
- Post Quantum Cryptography from Mutant Prime Knots [MP10]
- Public-key encrytion based on chebyshev maps [KT03]
- Public-key encryption with chaos [KSFV04]
- Paint and clocks

α	the power in a C^* construction
$\mathbf{a}, \mathbf{b}, \mathbf{c}$	constant vectors
$\mathbf{c}_S, \mathbf{c}_T$	constant parts of linear maps S, T
$C^* = (c_1^*, c_2^*, \ldots, c_n^*)$	the Matsumoto-Imai map $C^*_{q,n,\alpha} : \mathbf{x} \mapsto \mathbf{y} = \mathbf{x}^{q^\alpha+1}$ in $\mathbb{F}_{q^n}$
DF	(symmetric) differential of the function/map F
D, D_{reg}, and D_{XL}	degree in system-solving degree, operating degree of $\mathbf{F_4}$/$\mathbf{F_5}$ and XL
$\mathbb{F}_q$	finite (Galois) field of q elements, any representation of
g	sometimes, a generator of $\mathbb{K} = \mathbb{F}_q$
H_i	symmetric matrices for quadratic part of p_i (or z_i) in w_i
h, i, j, k, l	index variables, k often $:= [\mathbb{L} : \mathbb{K}]$, dimension of $\mathbb{L}$ over $\mathbb{K}$
$\mathcal{K}$	denoting a kernel
$\ker_{\mathbf{v}} f$	kernel of the symmetric matrix denoting quadratic part of f as function of $\mathbf{v}$.
$\mathbb{K}$	the base field, usually $= \mathbb{F}_q$
$\mathbb{L}$	$\mathbb{F}_{q^k}$, a field that is larger than $\mathbb{K}$
M_i	symmetric matrices for the quadratic part of y_i in x_j
M_S, M_T	matrices of linear maps S, T.
m	number of equations
$\mathtt{m}$	a multiplication, as a unit of time
n	number of variables
$O()$, $o()$, $\Omega()$	standard big-O, small-o, Omega notations
o	number of oil variables
P_{ik}	Matsumoto-Imai notation for coefficient of w_i in z_k
$\mathcal{P} = (p_1, \ldots, p_m)$	public map
Q_{ik}	Matsumoto-Imai notation for coefficient of w_i^2 in z_k
$\mathcal{Q} = (q_1, \ldots, q_m)$	central map
q	the size of the base field
R_{ijk}	Matsumoto-Imai notation for coefficient of $w_i w_j$ in z_k
R	$\lvert\mathcal{R}\rvert$, the number of relations (equations) in XL or $\mathbf{F_4}$
$\mathcal{R}^{(D)}$ or $\mathcal{R}$	Set of equations in XL or $\mathbf{F_4}$
r	usu. the minimum rank or # of removed (minus) equations
S	the initial linear map, $S(\mathbf{w}) = \mathbf{x} = \mathrm{M}_S\mathbf{w} + \mathbf{c}_S$
T	the final linear map, $T(\mathbf{y}) = \mathbf{z} = \mathrm{M}_T\mathbf{y} + \mathbf{c}_T$, or #terms in XL ($\lvert\mathcal{T}\rvert$ below)
$\mathcal{T}^{(D)}$ or $\mathcal{T}$	set of terms (monomials) in XL or $\mathbf{F_4}$
u	often the high rank parameter or # of Rainbow stages
v	number of vinegar variables
$v_1 < v_2 < \cdots < v_{u+1} = n$	structure of Rainbow $(v_1, o_1, \ldots, o_u)$, $o_i := v_{i+1} - v_i$
$\mathbf{w} = (w_1, \ldots, w_n)$	signature or plaintext block
X_i, Y_j	elements in intermediate fields
$\mathbf{x} = (x_1, \ldots, x_n)$	central variables, input to central map $\mathcal{Q}$
$\mathbf{y} = (y_1, \ldots, y_m)$	output of central map $\mathcal{Q}$, central polynomials
$\mathbf{z} = (z_1, \ldots, z_m)$	digest or ciphertext block

Table 1. Notations and Terminology

Lattices

- A lattice is just a set of points in $\mathbb{R}^n$ with a periodic structure.
- A basis of a vector space (like $\mathbb{R}^n$) is a set of vectors such that $\forall v \in V, v = \sum x_i b_i$ for $x_i \in \mathbb{R}$.
- A basis of a lattice is the same, but we require the coefficients to be integral ($x_i \in \mathbb{Z}$)

Lattice Hardness

- Lattice hardness problems are cool because they are worst case, rather than average case.
- SVP: Given a lattice basis, find the shortest nonzero vector in the lattice
- A few variants, most other lattice hardness problems can reduce to SVP

LLL

- The LLL algorithm solves SVP in polytime, so how can it be used as a hardness problem for crypto?
- SVP: Given a lattice basis, find the shortest nonzero vector in the lattice
- GapSVP: Given SV u, is $||u|| < 1$ or at most some β?
- Ajtai98 proves SVP is NP-Hard
- CVP: Given v (not necessarily in Λ), find $u \in \Lambda$ such $||u - v|| < ||y - v|| \quad \forall y \in \Lambda$
- Most other lattice problems, including CVP, reduce to the hardness of SVP

LLL

- The LLL algorithm approximates SVP in polytime up to factor $2^{(O(n))}$
- Not a problem since you can instantiate hard instances of SVP
- Most other lattice hardness problems can reduce to SVP like CVP.

Learning with errors

- A hardness problem from machine learning!
- Given $A \in \mathbb{Z}_q^{m \times n}$, $s, e \in \mathbb{Z}_q^m$, let $v = As + e$
- Given (A, v) find s.
- Most other lattice hardness problems can reduce to SVP like CVP.

A Quantum Reduction to SVP

- A (classical) efficient LWE algorithm implies an quantum efficient algorithm for solving SVP
- Given oracle access to solving CVP, You can uncompute a quantum state
- $|x\rangle \otimes |y\rangle \rightarrow |x\rangle \otimes |0\rangle$
- Motivation for lattice use in PQ crypto compared to other hardness assumptions.

Recall

- Every Field is a Ring
- A Ring does not require:
- Multiplicative inverses ($\forall a \exists b\ ab = 1_R$)
- A multiplicative identity ($\exists 1_R \forall a\ a1_R = 1_R a = a$)
- communitivity of multiplication ($\forall a, b\ ab \neq_? ba$)
- the two identites are different ($1_R \neq 0_R$)

Ring Learning with Errors (RLWE)

- $\mathbb{Z}/q\mathbb{Z} = \mathbb{Z}_q$ is a field.
- $\mathbb{Z}_q[x]$ is a ring of polynomials with coefficients from $\mathbb{Z}_q$
- $\Phi(x)$ is an irreducible polynomial from $\mathbb{Z}_p[x]$
- $\mathbb{Z}_p[x]/\Phi(x)$ is a quotient ring.
- Given sets $a_i(x), b_i(x) = a_i(x)s(x) + e_i(x)$, find $s_i(x), e_i(x)$
- Parameterization by $\Phi(x)$, degree n, and error bound b on s, e_i.

Ring Learning with Errors (RLWE)

- $\mathbb{Z}/q\mathbb{Z} = \mathbb{Z}_q$ is a field.
- $\mathbb{Z}_q[x]$ is a ring of polynomials with coefficients from $\mathbb{Z}_q$
- $\Phi(x)$ is an irreducible polynomial from $\mathbb{Z}_p[x]$
- $\mathbb{Z}_p[x]/\Phi(x)$ is a quotient ring.
- Given sets $a_i(x), b_i(x) = a_i(x)s(x) + e_i(x)$, find $s_i(x), e_i(x)$
- Parameterization by $\Phi(x)$, degree n, and error bound b on s, e_i.

GLYPH Signature Scheme Generation

- $s, e \xleftarrow{\$} \mathbb{Z}_p[x]/\Phi(x)$
- $t = as + e$
- s, e is the private key
- t is the public key

GLYPH Signature Scheme Signing

- $y_1, y_2 \xleftarrow{\$} \mathbb{Z}_p[x]/\Phi(x)$
- $w := ay_1 + y_2$
- $c := POLYHASH(\omega_w || m)$
- $z_1 := sc + y_1$
- $z_2 := ec + y_2$
- The signature is then c, z_1, z_2

GLYPH Signature Scheme Verification

- $w' := az_1 + z_2 - tc$
- $c' = POLYHASH(\omega_{w'}|m)$
- verify $c' = c$

GLYPH Signature Scheme Correctness

- $az_1 + z_2 - tc =$
- $a(sc + y_1) + z_2 - (as + e)c =$
- $asc + ay_1 + z_2 - asc - ec =$
- $ay_1 + z_2 - ec =$
- $ay_1 + (ec + y_2) - ec = ay_1 + y_2 = w$

LWE Relation to Codes

- Equivalently presented as decoding random linear codes
- Decisional vs Search
- Given $a_i(x), b_i(x)$, determine if $b_i = a_i(x)s(x) + e_i(x)$ or if $b_i(x)$ is drawn uniformly random
- If there exists such a distinguisher, then you can efficiently solve the McEliece Problem
- Given G, t find m such that $mG - c = t$

References

- https://cims.nyu.edu/ regev/papers/qcrypto.pdf
- Post Quantum Cryptography by D.J. Bernstein
- https://nvlpubs.nist.gov/nistpubs/ir/2019/NIST.IR.8240.pdf
- https://eprint.iacr.org/2017/766.pdf
- Given G, t find m such that $mG - c = t$