Ivy: A new IND-CCA-secure code-based public-key scheme

Li-Ping Wang
Institute of Information Engineering, CAS, China

The 5th Sino-Korea International Conference on Coding Theory and Its Related Topics
Shanghai, July 4, 2018

Outline

- Motivation
- Preliminaries
- IvyPKE: IND-CPA-secure public-key encryption
- IvyKEM: IND-CCA-secure key encapsulation mechanism
- Choice of parameters
- Conclusions

- Public-key cryptography has entered post-quantum era
  1. The first generation PKC: Integer-factoring-based RSA (1977)
  2. The second generation PKC: discrete-logarithm-based ECC (1984)
  3. Peter Shor's quantum algorithm (1994)

- The increased interest focuses on post-quantum cryptography recently due to
  1. Fast advance in quantum computers
  2. announcement by NIST to call for new standards for post-quantum cryptosystems

- Alternative public-key cryptosystems resistant to quantum computing attacks
  1. Lattice-based cryptography
  2. Code-based cryptography
  3. Multivariate-based cryptography

- There are 82 proposals submitted to NIST, in which 69 schemes enter the first round.
  1. 29 lattice-based schemes
  2. 20 code-based schemes
  3. 10 mulivaribles-based schemes
  4. 10 other-based schemes

- Announcement by Chinese Association for Cryptologic Research (CACR) to call for new cryptographic schemes including post-quantum schemes in June, 2018.
- submission deadline: Feb. 28, 2019

- $\mathbb{F}_{q^m}^n$: n-dimensional vector space over a finite field $\mathbb{F}_{q^m}$.
- $(\beta_1, \ldots, \beta_m)$: a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$.
- $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{F}_{q^m}^n$: $x_i = \sum_{j=1}^{m} x_{ji}\beta_j$, $1 \leq i \leq n$.
- $\bar{\mathbf{x}} = \begin{pmatrix} x_{11} & x_{12} & \ldots & x_{1n} \\ x_{21} & x_{22} & \ldots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \ldots & x_{mn} \end{pmatrix} \in \mathbb{F}_q^{m \times n}$.

- $\mathrm{Supp}(\mathbf{x}) :=< x_1, \ldots, x_n >_{\mathbb{F}_q}$; the $\mathbb{F}_q$-linear space of $\mathbb{F}_{q^m}$ spanned by the coordinates of $\mathbf{x}$.
- $w_R(\mathbf{x}) := rank(\bar{\mathbf{x}}) = \dim(Supp(\mathbf{x}))$.
- $\begin{bmatrix} m \\ r \end{bmatrix}_q = \prod_{i=0}^{r-1} \frac{q^m - q^i}{q^r - q^i} = \Theta(q^{r(m-r)})$: the number of supports of dimension r is the number of linear subspaces of $\mathbb{F}_{q^m}$ of dimension r.

For integers $1 \leq k \leq n$, a linear rank-metric code C of length n and dimension k over $\mathbb{F}_{q^m}$ is a subspace of dimension k of $\mathbb{F}_{q^m}^n$ embedded with the rank metric.
It is spanned by the rows of a matrix $G \in \mathbb{F}_q^{k \times n}$, called by generator matrix of C.
The $(n-k) \times n$ generator matrix H of $C^\perp$ is the parity check matrix of C.

q-linearized polynomials

- $L(x) = \sum_{i=0}^{d} a_i x^{q^i}, a_i \in \mathbb{F}_{q^m}, a_d \neq 0$: a q-linearized polynomial over $\mathbb{F}_{q^m}$, and d is called the q-degree of $f(x)$, denoted by $\deg_q(f(x))$.
- $\mathcal{L}_q(x, \mathbb{F}_{q^m})$: the set of all q-linearized polynomials over $\mathbb{F}_{q^m}$.
- multiplication operation: the composition $L_1(x) \circ L_2(x) = L_1(L_2(x))$.
- The set $\mathcal{L}_q(x, \mathbb{F}_{q^m})$ forms a non-commutative ring under the operations of composition $\circ$ and ordinary addition.

Gabidulin codes

Gabidulin codes were introduced by Gabidulin in 1985 and independently by Delsarte in 1978. They can be seen as the q-analog of Reed-Solomon codes. Let $g_1, \ldots, g_n \in \mathbb{F}_{q^m}$ be linearly independent over $\mathbb{F}_q$ and the Gabidulin code $\mathcal{G}$ is defined as follow:

$$\begin{aligned} \mathcal{G} \;=\; & \{(m(g_1), \ldots, m(g_n)) \in \mathbb{F}_{q^m}^n \,|\, m(x) \in \mathcal{L}_q(x, \mathbb{F}_{q^m}) \\ & \text{and } \deg_q(m(x)) < k\}. \end{aligned}$$

The Gabidulin code $\mathcal{G}$ with length n has dimension k over $\mathbb{F}_{q^m}$ and the generator matrix of $\mathcal{G}$ is as follows:

$$G = \begin{pmatrix} g_1 & \cdots & g_n \\ g_1^q & \cdots & g_n^q \\ \vdots & \ddots & \vdots \\ g_1^{q^{k-1}} & \cdots & g_n^{q^{k-1}} \end{pmatrix}.$$

The minimum rank distance of these codes is $n - k + 1$, and so they can efficiently decode up to $\frac{n-k}{2}$ errors.

Decoding of Gabidulin codes

The algorithm employed in order to decode Gabidulin codes was proposed by Sidorenko [1], etc, which is the generalization of Berlekamp-Massey algorithm and its complexity is $O(n^2)$.

[1] V. Sidorenko, G. Richter, M. Bossert. Linearized shift-register synthesis, IEEE Transactions on Information Theory, vol. 57, No. 9, 2011.

Difficult problems used in our cryptosystem

Rank syndrome decoding problem (RSD for short)

Given a check parity matrix $H \in \mathbb{F}_{q^m}^{(n-k)\times n}$ of a random linear code, and $\mathbf{y} \in \mathbb{F}_{q^m}^{1\times(n-k)}$, the goal is to find $\mathbf{x} \in \mathbb{F}_{q^m}^{1\times n}$ with $w_R(\mathbf{x}) = w$ such that $H\mathbf{x}^T = \mathbf{y}^T$.

The RSD problem has recently been proven difficult with a probabilistic reduction to the Hamming setting in [2]. As we all know, syndrome decoding problem in Hamming metric is NP-hard [3].

[2] Philippe Gaborit and Gilles Zemor. On the hardness of the decoding and the minimum distance problems for rank codes. IEEETrans. Information Theory, 62(12): 7245-7252, 2016.
[3] E. Berlekamp, R. McEliece and H. Van Tilborg. On the inherent intractability of certain coding problems, IEEE on IT, vol. 24, No. 3, 384-386, 1978.

Decisional version of RSD problem

Given input $(H, \mathbf{y}^T) \xleftarrow{\$} \mathbb{F}_{q^m}^{(n-k)\times n} \times \mathbb{F}_{q^m}^{(n-k)\times 1}$, the decision RSD problem asks to decide with non-negligible advantage whether $(H, \mathbf{y}^T)$ came from the RSD or the uniform distribution over $\mathbb{F}_{q^m}^{(n-k)\times n} \times \mathbb{F}_{q^m}^{(n-k)}$.

Security definiion

An Encryption scheme (Keygen, Enc, Dec) has to satisfy both Correctness and IND-CPA security properties.

Correctness: For every pair of keys (pk. sk) and every message m, we have $P[Dec(sk, Enc(pk, m, \theta)) = m] = 1 - negl(\lambda)$, for $negl(\cdot)$ a negligible function, λ is security parameter.

IND-CPA: indistinguishability under chosen plaintext attacks

A public-key encryption scheme $(KeyGen, Enc, Dec)$ is IND-CPA if any probabilistic polynomial-time adversary $\mathcal{A}$ can only succeed with probability at most $\frac{1}{2} + negl(\lambda)$ in the following experiment:

1. The challenge oracle runs Keygen(1^λ) to generate the random pair (pk,sk) and gives pk to $\mathcal{A}$;
2. $\mathcal{A}$ outputs two equal-length messages (m_0, m_1), and transmits them to a challenge oracle along with pk;
3. The challenger oracle selects a random $b \in \{0, 1\}$, and computes $c^* \leftarrow Enc(m_b, pk, \theta)$ and returns the ciphertext c^* to $\mathcal{A}$;
4. $\mathcal{A}$ outputs a guess b' for the value of b. If $b' = b$, we say $\mathcal{A}$ succeeds.

Note that the standard security requirement for a public key cryptosystem is IND-CCA2, i.e., indistinguishability against chosen ciphers attacks, and not just IND-CPA. The main difference is that for IND-CCA2 indistinguishability must hold even if the attacker is given a decryption oracle.

IND-CCA: indistinguishability against chosen ciphers attacks

A public-key encryption scheme $(KeyGen, Enc, Dec)$ is IND-CCA if any probabilistic polynomial-time adversary $\mathcal{A}$ can only succeed with probability at most $\frac{1}{2} + negl(\lambda)$ in the following experiment:

1. The challenge oracle runs Keygen(1^λ) to generate the random pair (pk,sk) and gives pk to $\mathcal{A}$;
2. $\mathcal{A}$ makes any polynomial number of queries on the cipher-texts he chooses to the decryption oracle $Dec(sk, \cdot)$;
3. $\mathcal{A}$ outputs two equal-length messages (m_0, m_1), and transmits them to a challenge oracle along with pk;
4. The challenger oracle selects a random $b \in \{0, 1\}$, and computes $c^* \leftarrow Enc(m_b, pk, \theta)$ and returns the ciphertext c^* to $\mathcal{A}$;
5. There are two cases:
   - In IND-CCA1, the adversary may not make the further calls to the decryption oracle.
   - In IND-CCA2, the adversary may make further calls to the decryption oracle, but may not submit the challenge ciphertext C^*.
6. $\mathcal{A}$ output a guess b' for the value of b. If $b' = b$, we say $\mathcal{A}$ succeeds.

In [4, 5], a generic transform is presented to pass an IND-CPA encryption scheme into an IND-CCA2 KEM.

[4] A. Fujisaki and T. Okamoto, Secure integration of asymmetric and symmetric encryption schemes, CRYPTO'99, LNCS Vol. 1666, Springer, Heidelberg, 537-554

[5] D. Hofheinz, K. Hovelmanns, and E. Kiltz. A modular analysis of the fujisaki-okamoto transformation. Cryptology ePrint Archive, Report 2017/604, 2017.

Key encapsulation mechanism KEM

KEMs are a class of encryption techniques designed to secure symmetric cryptographic key material for transmission using asymmetric public-key algorithms.

KEMs simplify the traditional key exchange process by using hash function instead of padding.

The key exchange protocol, or authenticated key exchange protocol, etc, can be obtained by a direct application of the KEM.

An example

Traditional way: Bob first turns M into a larger integer $1 \leq m \leq n$ by using an agreed-upon reversible protocol known as padding scheme, such as OAEP. He computes the ciphertext $c = m^e \mod n$.
Alice can recover m from c by $m = c^d \mod n$.
Given m, she recovers the original message M by reversing the padding scheme.

An example -continued

KEM way: Bod generates a random m, $1 \leq m \leq n$.
He derives his key M by $M = KDF(m)$, where KDF is a key derivation function, such as cryptographic hash. He then computes the ciphertext $c = m^e \mod n$.
Alice then recovers m by $m = c^d \mod n$.
Given m, she can recover the key M by $M = KDF(m)$.

Key Generation Ivy.PKE.Keygen(): $\rho, \sigma \xleftarrow{\$} \{0,1\}^{256}$, $H \sim \mathbb{F}_{q^m}^{n \times n} :=$ Shake256(ρ) and $X, Y \sim \mathbb{F}_{q^m}^{n \times n} :=$ Shake256(σ) with $w_R(X_i) = w_R(Y_i) = w_e$, where X_i and Y_i denote the ith column of X and Y. Set $Q := HX + Y$.
The public key: $pk = (\rho, Q)$. The secret key: $sk = X$.

Encryption Ivy.PKE.Enc(pk$=(\rho, Q), m \in \mathcal{M}$):
Let $G \in \mathbb{F}_{q^m}^{k \times n}$ be the generator matrix of a Gabidulin code $\mathcal{G}$. Set $\gamma \xleftarrow{\$} \{0,1\}^{256}$ and $\mathbf{r}, \mathbf{e}_1, \mathbf{e}_2 \sim \mathbb{F}_{q^m}^{1 \times n} :=$ Shake256(γ) with $w_R(\mathbf{e}_1) = w_R(\mathbf{e}_2) = w_R(\mathbf{r}) = w$. Return the ciphertext $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2) \in \mathbb{F}_{q^m}^{1 \times 2n}$, where

$$\mathbf{c}_1 = \mathbf{r}H + \mathbf{e}_1, \mathbf{c}_2 = \mathbf{r}Q + \mathbf{e}_2 + mG.$$

Decryption Ivy.PKE.Dec(sk,$\mathbf{c}$)
Compute $\mathbf{c}_2 - \mathbf{c}_1 X = mG + \mathbf{r}Y + \mathbf{e}_2 - \mathbf{e}_1 X$. Since $w_R(\mathbf{r}Y + \mathbf{e}_2 - \mathbf{e}_1 X) = w_e w_r \leq \frac{n-k}{2}$, then m is obtained by decoding algorithm of $\mathcal{G}$.

Correctness.

The correctness of our new encryption scheme clearly relies on the decoding capability of the code $\mathcal{G}$. Specifically, assuming that $\mathcal{G}$. decode correctly decodes $\mathbf{c}_2 - \mathbf{c}_1 X$, we have

$$Decrypt(sk, Encrypt(m, pk)) = m.$$

And $\mathcal{G}$.decode correctly decodes $w_R(\mathbf{r}Y + \mathbf{e}_2 - \mathbf{e}_1 X)$ whenever $w_R(\mathbf{r}Y + \mathbf{e}_2 - \mathbf{e}_1 X) \leq \delta = \frac{n-k}{2}$.

Let us check the error distribution.
Suppose that X_i and Y_i are taken from the same vector space with dimension w_e and the basis is $\{\alpha_1, \ldots, \alpha_{w_e}\}$ including 1. Similarly, suppose that $\mathbf{e}_1$, $\mathbf{e}_2$ and $\mathbf{r}$ are taken from the vector space with dimension w_r and the basis is $\{\beta_1, \ldots, \beta_{w_r}\}$.
Therefore, the dimension of the subspace spanned by $\mathbf{r}Y + \mathbf{e}_2 - \mathbf{e}_1 X$ is at most $w_e w_r$.

For decoding, we consider Gabidulin code $[k, n]$ over $\mathbb{F}_{q^m}$, which can decode $\frac{n-k}{2}$ rank errors and choose our parameters such that $w_e w_r \leq \frac{n-k}{2}$, so that there is no decryption failure.

Theorem

The encryption scheme IvyPKE is IND-CPA secure under the RSD assumption.

Sketch of proof. Due to decision RSD problem, the pair $(H, Q = HX + Y)$ is indistinguishable from (H, T), where $X, Y, T \in \mathbb{F}_{q^m}^{n \times n}$ are chosen randomly. In addition, $(H, \mathbf{r}H + \mathbf{e}_1)$ and $(Q, \mathbf{r}Q + \mathbf{e}_2)$ are distinguishable from $(H, \mathbf{t}_1)$ and $(Q, \mathbf{t}_2)$, where $\mathbf{e}_1, \mathbf{e}_2, \mathbf{t}_1, \mathbf{t}_2, \mathbf{r} \in \mathbb{F}_{q^m}^{n}$ are chosen randomly.

Key Generation Ivy.KEM.Keygen():
It is the same as the Ivy.PKE.Keygen().

Encapsulation Ivy.KEM.Encaps(pk=(ρ, Q)):
1. $\mathbf{m} \leftarrow \{0,1\}^{256}$
2. $(K, d, \mathbf{r}) := G(pk, \mathbf{m})$
3. $\mathbf{c} :=$ Ivy.PKE.Enc$(\rho, Q, \mathbf{m}; \mathbf{r})$
4. $ss := H(\mathbf{c} \parallel K \parallel d)$
5. return$(\mathbf{c}, d; ss)$

Decapsulation Ivy.KEM.Decaps$(sk = (X, \rho, Q), \mathbf{c})$
1. Compute $m' :=$ Ivy.PKE.Dec$(X, \mathbf{c})$
2. $(K', d', \mathbf{r}') := G(pk, \mathbf{m}')$
3. $\mathbf{c}' :=$ Ivy.PKE.Enc$(\rho, Q, \mathbf{m}'; \mathbf{r}')$
4. if $\mathbf{c}' = \mathbf{c}$ and $d = d'$ then
5. return $ss := H(\mathbf{c}' \parallel K' \parallel d')$
6. else the decapsulation fails and return $\perp$
7. end if

Theorem IvyKEM scheme is IND-CCA2 secure under the RSD assumption provided that G, H are random oracles.

Table: Parameter sets for IvyKEM

instance	q	m	n	k	w	w_r	Security
Ivy-I	2	89	64	4	5	6	128
Ivy-II	2	110	90	6	6	7	192
Ivy-III	2	130	100	4	6	8	256

Table: The theoretical sizes in byte for IvyKEM

Instance	pk size	sk size	ct size	ss size	Security
Ivy-I	$45,600$	$45,600$	1488	64	128
Ivy-II	$111,407$	$111,407$	2539	64	192
Ivy-III	$162,532$	$162,532$	3314	64	256

Table: The theoretical sizes for Classic McEliece

Instance	q	m	n	k	t	w	security	pk size
Classic	2	13	8192	7815	29	29	128	$368,282$B
McEliece	2	13	6960	5413	119	119	256	$1,046,737$B

Table: The theoretical sizes in byte for RQC

Instance	pk size	sk size	ct size	ss size	Security
RQC-I	1491	1491	1555	64	128
RQC-II	2741	2741	2805	64	192
RQC-III	3510	3510	3574	64	256

Generic Attacks

The attack to the IND-CCA2 security of IvyKEM is to solve the rank syndrome decoding problem with parameters (q^m, n, k, w, w_e) in Table 1.

For an $[n, k]$ rank code over $\mathbb{F}_{q^m}$, the best combinatorial attacks to decode a word with errors of weight w is

$$O((nm)^3 q^{r\left\lceil \frac{m(k+1)}{n} \right\rceil - m}).$$

The main contribution in this talk is that we propose a semantically secure public-key encryption scheme whose security is based on rank syndrome decoding problem. Then applying a variant of the Fujisaki-Okamoto transform, we give an IND-CCA-secure KEM.

Security.

- Our design rational is that security is first and cost is second.
- Our scheme is based on the hardness of rank syndrome decoding problem, which is proved to be NP-hard.
- It cannot be excluded that some fatal attacks are possible in the future since existing all code-based scheme are of certain structures.
- It is very worth that we propose such a provably secure code-based cryptosystem.

Efficiency. Although the public-key size is much bigger than these in RQC, etc, it is much shorter than classic McEliece.

Similarity with LWE. Our proposal is very similar to LWE problem. FrodoKEM and LOTUS are based on LWE problem and are NIST candidates.

No decoding failure.

Thank You for Your Attention!