Zeng *et al. Hum. Cent. Comput. Inf. Sci.* (2019) 9:32
https://doi.org/10.1186/s13673-019-0193-6

Human-centric Computing and Information Sciences

RESEARCH Open Access

An IND-CCA2 secure post-quantum encryption scheme and a secure cloud storage use case

Peng Zeng[1], Siyuan Chen[2*] and Kim-Kwang Raymond Choo[3]

*Correspondence: siyuanchen1900@outlook.com
[2] College of Computer Science and Technology, Shanghai University of Electric Power, Shanghai, China
Full list of author information is available at the end of the article

Abstract

Code-based public key encryption (PKE) is a popular choice to achieve post-quantum security, partly due to its capability to achieve fast encryption/decryption. However, code-based PKE has larger ciphertext and public key sizes in comparison to conventional PKE schemes (e.g., those based on RSA). In 2018, Lau and Tan proposed a new rank metric code-based PKE scheme, which has smaller public key and ciphertext sizes compared to other code-based PKE schemes. They also proved that their scheme achieves IND-CPA security, assuming the intractability of the decisional rank syndrome decoding problem. It is known that IND-CCA2 security is the strongest and most popular security assurance for PKE schemes. Therefore, in this paper, we obtain a new code-based PKE scheme from Lau and Tan's scheme, in order to inherit the underlying small public key and ciphertext sizes. However, our new scheme is shown to achieve IND-CCA2 security, instead of the weaker IND-CPA security. Specifically, the respective public key size and ciphertext size in our new scheme are 15.06 KB and 1.37 KB under 141-bit security level, and 16.76 KB and 1.76 KB under 154-bit security level. We then present a use case for the proposed scheme, that is for secure cloud storage.

Keywords: Post-quantum cryptography, Public key encryption, Rank metric codes, IND-CCA2 security, Cloud storage

Introduction

With rapid advances in Internet and information and communication technologies (ICT; e.g., computation devices, speed of communication and device processors), our society is becoming increasingly reliant on technologies. This is particularly true for technologically advanced countries. This has also given rise to a significant increase in the amount of data generated, and needed to be processed and stored (e.g., using distributed storage servers such as cloud server). Cloud storage service providers such as Amazon's S3 offer users public cloud storage services, where users can backup, access and/or process their data anywhere from some computing devices (e.g., Android and iOS devices) connected to the Internet. Popularity of such services is also driven by financial costs, as it may be cheaper to pay for what you use than building, maintaining and securing one's local storage services. One downside, however, is the potential data privacy and integrity risks since data is now outsourced to some public cloud service providers that may be

semi-trusted. Thus, massive techniques are proposed for ensuring the data security and decreasing the useless data such as semi-supervised learning, spammer detection framework and cryptographic protocol [1–4].

Generally, a cloud storage system should ensure availability (e.g., for any legitimate customer, (s)he can reach his/her uploaded data from some Internet-connected devices), reliability (of user data outsourced to the cloud), efficient retrieval (of user data outsourced to the cloud), data sharing (between authorized users), security (i.e., both confidentiality and integrity [5]), and other features required/stated in the service level agreements (SLAs). Cryptographic schemes, such as proxy re-encryption, attribute-based encryption, searchable encryption [6–9], have been designed to achieve several of these features.

Nowadays, most existing cryptographic schemes are number-theoretic-based [10–12], which are not resilient against Shor's quantum attack algorithm [13, 14] using quantum computers. Code-based public key encryption (PKE) proposed by McEliece [15] in 1978 is widely considered to be post-quantum secure [16]. There are two key advantages in McEliece-like code-based PKE schemes, namely: encryption/decryption speed is very fast, and its security relies on hard problems in coding theory which is believed to resist quantum computing attacks. However, such schemes are not widely used in practice, partly due to large public key size (in comparison to classic number-theoretic-based schemes). Hence, how to design a code-based PKE scheme with small public key size is an area of active research. In 2017, for example, Loidreau proposed a code-based PKE scheme from rank metric [17], which can significantly reduce the size of public keys compared to Hamming metric encryption schemes. More recently in 2018, Lau and Tan introduced a public key generation approach and a new rank metric code-based PKE scheme (hereafter refer to LT scheme) to further reduce the public key size [18]. The LT scheme was also shown to be indistinguishability against chosen plaintext attacks (IND-CPA) secure, based on the decisional rank syndrome decoding (DRSD) assumption.

Our contribution

For a PKE scheme, however, the strongest and the most popular security notion is indistinguishability against adaptive chosen ciphertext attacks (IND-CCA2). This is the gap we seek to address in this paper. Our contribution in this paper include three aspects. Firstly, we propose a new IND-CCA2 secure PKE scheme based on the LT scheme. While the semantic security in our PKE scheme has been improved to be IND-CCA2 secure, the public key and ciphertext sizes remains small. Secondly, we give the formal proof of our PKE scheme under the DRSD assumption. Thirdly, we present a use case of our PKE scheme in secure cloud storage.

Paper organization

The rest of the paper is organized as follows. Prior to presenting our scheme, we introduce relevant background materials in "Preliminaries" section and the LT scheme [18] in "Revisiting the LT scheme" section. We then present our IND-CCA2 secure code-based PKE scheme in "Our IND-CCA2 secure PKE scheme" section, following by its security proof and performance evaluation in "Security proof" and "Efficiency" sections. For example, we show that the public key size in our new scheme is 15.06 KB

under 141-bit security level which is acceptable in many today's applications. We use cloud storage as a use case and explain its potential utility in "A secure cloud storage use case" section. Finally, we conclude this paper in "Conclusion" section.

Preliminaries

In this section, we revisit the definitions, notations and materials relevant to rank metric codes, which is required in the understanding of our proposed scheme. In the rest of this paper, we will use the notations and the functions as we defined in Table 1.

Definition 1 (*Linear code and its generator matrix*) An $[n,k]_q$ linear code $\mathcal{C}$ is a linear subspace of $\mathbb{F}_q^n$ with dimension k, and a matrix $G \in \mathbb{F}_q^{k\times n}$ is called a generator matrix of $\mathcal{C}$ if its row vectors form a basic of $\mathcal{C}$.

Definition 2 (*Rank metric*) Let $\mathbb{F}_{q^m}$ be an extension field of $\mathbb{F}_q$ for some positive integer m, and $\beta = (\beta_0, \beta_1, \ldots, \beta_{m-1})$ a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. For each vector $\mathbf{a} = (a_0, a_1, \ldots, a_{n-1}) \in \mathbb{F}_{q^m}^n$, we associate it with an $m \times n$ matrix $M(\mathbf{a}) = (a_{ij})_{0\le i\le m-1, 0\le j\le n-1} \in \mathbb{F}_q^{m\times n}$ s.t. $a_j = \sum_{i=0}^{m-1} m_{ij}\beta_i$, $j = 0, 1, \ldots, n-1$. The rank of $\mathbf{a}$, denoted by $\|\mathbf{a}\|$, is defined as the rank of the matrix $M(\mathbf{a})$, and the rank distance between any two vectors $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{F}_{q^m}^n$ is defined by $\|\mathbf{x}-\mathbf{y}\|$.

Definition 3 (*Circulant matrix*) Given a vector $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1}) \in \mathbb{F}_{q^m}^n$, we associate it with an $n \times n$ circulant matrix $Rot(\mathbf{c}) = (c_{ij})_{0\le i,j\le n-1} \in \mathbb{F}_{q^m}^{n\times n}$ satisfying $c_{ij} = c_{(i-j) \mod n}$ for $0 \le i \le n-1, 0 \le j \le n-1$. That is, we have

$$Rot(\mathbf{c}) = \begin{pmatrix} c_0 & c_{n-1} & \cdots & c_2 & c_1 \\ c_1 & c_0 & c_{n-1} & c_3 & c_2 \\ \vdots & c_1 & c_0 & \ddots & \vdots \\ c_{n-2} & \vdots & \ddots & \ddots & c_{n-1} \\ c_{n-1} & c_{n-2} & \cdots & c_1 & c_0 \end{pmatrix}.$$

Table 1 Some notations in this paper

$\mathbb{F}_q$	A finite field of q elements
$\|$	The concatenation operator of vectors or matrices
$x \xleftarrow{\$} S$	The operation of selecting an element x from set A uniformly at random
$\|\|\mathbf{a}\|\|$	The rank of a vector $\mathbf{a}$
$Len(\mathbf{x})$	A function that takes as input a vector $\mathbf{x}$, and outputs the length of $\mathbf{x}$
$Rd_k(\mathbf{x})$	A function that takes as input a seed $\mathbf{x}$, and outputs a random vector from $\mathbb{F}_{q^m}^k$
$Conv_t(\mathbf{x})$	A function that takes as input a vector $\mathbf{x}$ of length $\lfloor \log_{q^m} \binom{n}{t} \rfloor$ over $\mathbb{F}_{q^m}$, and outputs an error vector of length n and weight t. The inverse of function $Conv_t(\cdot)$ is denoted by $Conv_t^{-1}(\cdot)$
$Lt_k(\mathbf{x})$	A function that takes as input a vector $\mathbf{x}$, and outputs the left k components
$Rt_k(\mathbf{x})$	A function that takes as input a vector $\mathbf{x}$, and outputs the right k components

Further, we use the notation $Rot_k(\mathbf{c})$ to denote a $k \times n$ matrix consisting of the first k rows of $Rot(\mathbf{c})$.

Definition 4 (*Decisional rank syndrome decoding (DRSD) problem*) For a given full rank matrix $A \in \mathbb{F}_{q^m}^{k \times n}$ and an integer t, we define $\mathcal{S}_{A,t,0} = \{(A, \mathbf{y}) \mid \mathbf{y} \xleftarrow{\$} \mathbb{F}_{q^m}^n\}$ and $\mathcal{S}_{A,t,1} = \{(A, \mathbf{xA} + \mathbf{e}) \mid \mathbf{x} \in \mathbb{F}_{q^m}^k, \mathbf{e} \in \mathbb{F}_{q^m}^n \text{ and } \|\mathbf{e}\| = t\}$.

Input: a full rank matrix $A \in \mathbb{F}_{q^m}^{k \times n}$, an integer t, and a vector $\mathbf{s} \in \mathbb{F}_{q^m}^n$.

Output: a bit $b \in \{0, 1\}$ s.t. $(A, \mathbf{s}) \in \mathcal{S}_{A,t,b}$.

Let $\mathcal{A}$ be a probabilistic polynomial time (PPT) algorithm for the DRSD problem. With respect to a security parameter λ, we define the advantage of $\mathcal{A}$ as

$$\mathrm{Adv}_{\mathcal{A}}^{\mathrm{DRSD}}(\lambda) = \left| \Pr[\mathcal{A}(A, t, \mathbf{s}) = b \mid b \xleftarrow{\$} \{0, 1\}, (A, \mathbf{s}) \xleftarrow{\$} \mathcal{S}_{A,t,b}] - \frac{1}{2} \right|.$$

In [19], the DRSD problem is proven to be NP hard in the worst case; thus, it is reasonable to assume that $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{DRSD}}(\lambda)$ is negligible with regard to λ.

Definition 5 (*IND-CPA attack*) To describe the IND-CPA attack we first introduce the power of the adversary, denoted by $\mathcal{A}$, in the attack.

Chosen Plaintext Adversary: $\mathcal{A}$ can choose the plaintext adaptively and he can get the corresponding ciphertext of the chosen plaintext through the encryption mechanism.

We define an IND-CPA attack as an experiment. The experiment runs the key generation algorithm and gets a public-private key pair (pk, sk). An IND-CPA adversary uses pk to get several message-ciphertext pairs and stores them in a table. A distinguisher $\mathcal{D}$ gets pk and the message-ciphertext table. $\mathcal{D}$ randomly chooses two messages m_0 and m_1 (not in the message-ciphertext table) and sends them to the encryption mechanism. The encryption mechanism randomly chooses a message m_b, $b \in \{0, 1\}$, and sends the corresponding ciphertext c_b to $\mathcal{D}$. $\mathcal{D}$ takes as input (pk, m_0, m_1, c_b) and outputs a guess b' of b. If $b' = b$, the experiment successes. Otherwise, the experiment failures. The probability that $\mathcal{D}$ succeeds in the IND-CPA experiment is denoted by $\Pr_{CPA}$.

A PKE scheme is called IND-CPA secure if $|\Pr_{CPA} - \frac{1}{2}|$ is negligible.

Revisiting the LT scheme [18]

The LT scheme [18] consists of four PPT algorithms, namely: Setup, KGen, Enc, and Dec which are described as follows.

- Setup(1^λ): Given a security parameter λ, the algorithm chooses a prime power q, a group of integers (n, m, k, k_0, k_1, t) s.t. $n > k$, $k_0 = \lfloor \frac{k}{2} \rfloor$, $k_1 = k - k_0, t \leq \lfloor \frac{n-k}{2} \rfloor$ according to λ. The algorithm then outputs the public parameter $params = (q, n, m, k, k_0, k_1, t)$.
- KGen($params$): Give the public parameter $params$, the algorithm

Enc$(pk, \mathbf{m} \in \mathbb{F}_{q^m}^{k_1})$:
STEP 1: $\mathbf{c}_0 = (\mathbf{m}_r \vert \mathbf{m}) Rot_k(\mathbf{x}) + \mathbf{e}_0$, where $\mathbf{m}_r \xleftarrow{\$} \mathbb{F}_{q^m}^{k_0}$ and $\mathbf{e}_0 \xleftarrow{\$} \mathbb{F}_{q^m}^{n}$ with $\Vert\mathbf{e}_0\Vert \leq \lfloor \frac{t}{2} \rfloor$;
STEP 2: Checks whether the equation $\Vert(\mathbf{m}_r \vert \mathbf{m}) Rot_k(\mathbf{x})\| > \lceil \frac{3}{4}(n-k) \rceil$ holds. If not, return to **STEP 1**;
STEP 3: $\mathbf{c}_1 = (\mathbf{m}_r \vert \mathbf{m})\hat{G} + \mathbf{e}_1$, where $\mathbf{e}_1 \xleftarrow{\$} \mathbb{F}_{q^m}^{n}$ with $\Vert\mathbf{e}_1\Vert \leq \lfloor \frac{t}{2} \rfloor$;
STEP 4: Outputs $\mathbf{c} = (\mathbf{c}_0, \mathbf{c}_1)$.

Fig. 1 Encryption algorithm in LT scheme

Dec$(sk, \mathbf{c})$:
STEP 1: $\mathbf{c}_1 - \mathbf{c}_0 P$;
STEP 2: Executes $\mathcal{DEC}(\mathbf{c}_1 - \mathbf{c}_0 P)$;
STEP 3: Outputs the right k_1 components of $\mathcal{DEC}(\mathbf{c}_1 - \mathbf{c}_0 P)Q^{-1}$.

Fig. 2 Decryption algorithm in LT scheme

- Chooses a generator matrix $G \in \mathbb{F}_{q^m}^{k \times n}$ of a $[n, k]_{q^m}$ linear code $\mathcal{C}$ with error correcting ability t and an efficient decoding algorithm $\mathcal{DEC}(\cdot)$;
- Chooses a vector $\mathbf{x} \xleftarrow{\$} \mathbb{F}_{q^m}^{n}$ s.t. $\Vert\mathbf{x}\Vert = n$;
- Chooses two invertible matrices $Q \in \mathbb{F}_{q^m}^{k \times k}$ and $P \in \mathbb{F}_{q}^{n \times n}$ uniformly at random;
- Computes $\hat{G} = QG + Rot_k(\mathbf{x})P$.

The public/private key pair is $(pk = (\hat{G}, \mathbf{x}), sk = (Q, G, P, \mathcal{DEC}(\cdot)))$.

The algorithms Enc and Dec are listed in Figs. 1 and 2, respectively.

As mentioned previously, the LT scheme was proven to be IND-CPA secure under the DRSD assumption in [18].

Our IND-CCA2 secure PKE scheme

In this section, we convert the LT scheme into an IND-CCA2 secure PKE scheme in which two error vectors come from a random value. This allows the new scheme to avoid the expansion of ciphertext.

Our new IND-CCA2 secure PKE scheme consists of the following four algorithms:

- Setup(1^{λ}): Given a security parameter λ, the algorithm
  - Chooses a prime number power q, a group of integers $(n, m, k, k_0, k_1, t, t_0, t_1)$ s.t. $n > k$, $k_0 = \lfloor \frac{k}{2} \rfloor$, $k_1 = k - k_0$, $t \leq \lfloor \frac{n-k}{2} \rfloor$, $t_0 = \lfloor \frac{t}{2} \rfloor$, $t_1 = t - t_0$;
  - Chooses a random vector $\mathbf{con}$ over $\mathbb{F}_{q^m}$ with $Len(\mathbf{con}) \leq \frac{1}{2}\ell$, where $\ell_0 = \lfloor \log_{q^m} \binom{n}{t_0} \rfloor$, $\ell_1 = \lfloor \log_{q^m} \binom{n}{t_1} \rfloor$, and $\ell = \ell_0 + \ell_1$;
  - Outputs the public parameter $params = (q, n, m, k, k_0, k_1, t, t_0, t_1, \mathbf{con})$.
- KGen$(params)$: Give the public parameter $params = (q, n, m, k, k_0, k_1, t, t_0, t_1, \mathbf{con})$, the algorithm

Enc$(pk, \mathbf{m} \in \mathbb{F}_{q^m}^{k_1})$:

STEP 1: Generates a vector $\mathbf{v} \xleftarrow{\$} \mathbb{F}_{q^m}^{\ell - Len(\mathbf{con})}$;
STEP 2: $\mathbf{u}_1 = Rd_{k_1+Len(\mathbf{con})}(\mathbf{v}) + (\mathbf{m}|\mathbf{con})$;
STEP 3: $\mathbf{u}_2 = \mathbf{v} + \mathcal{H}(\mathbf{u}_1)$, where $\mathcal{H} : \{0,1\}^* \rightarrow \mathbb{F}_{q^m}^{\ell - Len(\mathbf{con})}$ is a secure hash function;
STEP 4: $\mathbf{u}_3|\mathbf{u}_4 = \mathbf{u}_1|\mathbf{u}_2$, where $Len(\mathbf{u}_3) = \ell$, $Len(\mathbf{u}_4) = k_1$;
STEP 5: $\mathbf{e}_0 = Conv_{t_0}(Lt_{\ell_0}(\mathbf{u}_3))$,
$\mathbf{e}_1 = Conv_{t_1}(Rt_{\ell_1}(\mathbf{u}_3))$;
STEP 6: $\mathbf{c}_0 = (\mathbf{m}_r|\mathbf{u}_4)Rot_k(\mathbf{x}) + \mathbf{e}_0$, where $\mathbf{m}_r \xleftarrow{\$} \mathbb{F}_{q^m}^{k_0}$;
STEP 7: Checks whether the equation $\|(\mathbf{m}_r|\mathbf{u}_4)Rot_k(\mathbf{x})\| > \lceil \frac{3}{4}(n-k) \rceil$ holds.
If not, returns to **STEP 6**.
STEP 8: $\mathbf{c}_1 = (\mathbf{m}_r|\mathbf{u}_4)\hat{G} + \mathbf{e}_1$;
STEP 9: Outputs $\mathbf{c} = (\mathbf{c}_0, \mathbf{c}_1)$.

Fig. 3 Encryption algorithm in our scheme

Dec$(sk, \mathbf{c})$:

STEP 1: $\mathbf{c}_1 - \mathbf{c}_0 P$;
STEP 2: $\mathcal{DEC}(\mathbf{c}_1 - \mathbf{c}_0 P)$;
STEP 3: $\mathbf{u}_4' = Rt_{k_1}(\mathcal{DEC}(\mathbf{c}_1 - \mathbf{c}_0 P)Q^{-1})$;
STEP 4: $\mathbf{e}_0' = \mathbf{c}_0 - (\mathcal{DEC}(\mathbf{c}_1 - \mathbf{c}_0 P)Q^{-1})Rot_k(\mathbf{x})$,
$\mathbf{e}_1' = \mathbf{c}_1 - (\mathcal{DEC}(\mathbf{c}_1 - \mathbf{c}_0 P)Q^{-1})\hat{G}$;
STEP 5: $Conv_{t_0}^{-1}(\mathbf{e}_0')$, $Conv_{t_1}^{-1}(\mathbf{e}_1')$ to get $\mathbf{u}_3' = Conv_{t_0}^{-1}(\mathbf{e}_0')|Conv_{t_1}^{-1}(\mathbf{e}_1')$;
STEP 6: $\mathbf{u}_1'|\mathbf{u}_2' = \mathbf{u}_3'|\mathbf{u}_4'$ s.t. $Len(\mathbf{u}_1') = k_1 + Len(\mathbf{con})$ and $Len(\mathbf{u}_2') = \ell - Len(\mathbf{con})$;
STEP 7: Outputs $\mathbf{m}'$ if $\mathbf{con}' = \mathbf{con}$ and $\perp$ otherwise.

Fig. 4 Decryption algorithm in our scheme

- Chooses a generator matrix $G \in \mathbb{F}_{q^m}^{k \times n}$ of an $[n,k]_{q^m}$ linear code $\mathcal{C}$ with error correcting ability t and an efficient decoding algorithm $\mathcal{DEC}(\cdot)$;
- Chooses a vector $\mathbf{x} \xleftarrow{\$} \mathbb{F}_{q^m}^n$ s.t. $\|\mathbf{x}\| = n$;
- Chooses two invertible matrices $Q \in \mathbb{F}_{q^m}^{k \times k}$ and $P \in \mathbb{F}_q^{n \times n}$ uniformly at random;
- computes $\hat{G} = QG + Rot_k(\mathbf{x})P$.

The public key is $pk = (\hat{G}, \mathbf{x})$ and the private key is $sk = (Q, G, P, \mathcal{DEC}(\cdot))$.

Similar to in "Revisiting the LT scheme" section, we list the algorithms Enc and Dec of our proposed PKE scheme in Figs. 3 and 4, respectively.

Security proof

In this section, we prove the correctness and IND-CCA2 security of our code-based PKE scheme presented in the preceding section.

Correctness

According to the Enc and Dec algorithms of our scheme, we can easily get:

$$\begin{aligned}\mathbf{u}_4' &= Rt_{k_1}(\mathcal{DEC}(\mathbf{c}_1 - \mathbf{c}_0 P)Q^{-1}) \\ &= Rt_{k_1}(\mathcal{DEC}((\mathbf{m}_r|\mathbf{u}_4)\hat{G} + \mathbf{e}_1 - (\mathbf{m}_r|\mathbf{u}_4)Rot_k(\mathbf{x})P - \mathbf{e}_0 P)Q^{-1}) \\ &= Rt_{k_1}(\mathcal{DEC}((\mathbf{m}_r|\mathbf{u}_4)QG + \mathbf{e}_1 - \mathbf{e}_0 P)Q^{-1}) \\ &= Rt_{k_1}((\mathbf{m}_r|\mathbf{u}_4)QQ^{-1}) \\ &= \mathbf{u}_4.\end{aligned}$$

Then, we can compute the error vectors $\mathbf{e}_0$ and $\mathbf{e}_1$ correctly with the vector $\mathbf{u}_4$ and get $\mathbf{u}_3' = Conv_{t_0}^{-1}(\mathbf{e}_0)|Conv_{t_1}^{-1}(\mathbf{e}_1) = \mathbf{u}_3$. From $\mathbf{u}_3$ and $\mathbf{u}_4$, we can recover $\mathbf{u}_1$ and $\mathbf{u}_2$ correctly and obtain the message $\mathbf{m}$ by computing

$$\mathbf{v} = \mathbf{u}_2 - \mathcal{H}(\mathbf{u}_1) \quad \text{and} \quad \mathbf{m}|\mathbf{con} = \mathbf{u}_1 - Rd_{k_1+Len(\mathbf{con})}(\mathbf{v}).$$

This concludes the correctness proof of our scheme.

IND-CCA2 security

The IND-CCA2 security for any PKE scheme can be defined as follows.

Definition 6 (*IND-CCA2 security*) For any two-stage (find stage and guess stage) adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against a PKE scheme $\mathcal{E} = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$, the IND-CCA2 security is modeled as an experiment $\mathbf{Exp}_{\mathcal{A},\mathsf{PKE}}^{\mathsf{IND\text{-}CCA2}}(\lambda)$ with respect to a security parameter λ.

$\mathbf{Exp}_{\mathcal{A},\mathsf{PKE}}^{\mathsf{IND\text{-}CCA2}}(\lambda)$:
$params \leftarrow \mathsf{Setup}(\lambda)$
$(pk, sk) \leftarrow \mathsf{KGen}(params)$
$(\mathbf{m}_0, \mathbf{m}_1,$ state $) \leftarrow \mathcal{A}_1^{\mathsf{Dec}(sk,\cdot)}(pk)$, where $\mathbf{m}_0$ and $\mathbf{m}_1$ have the same length
$b \xleftarrow{\$} \{0, 1\}$
$\mathbf{c}' \leftarrow \mathsf{Enc}(pk, \mathbf{m}_b)$
$b' \leftarrow \mathcal{A}_2^{\mathsf{Dec}(sk,\cdot)}(\mathbf{c}',$ state $)$. Note that $\mathcal{A}_2$ is not allowed to make decryption query on $\mathbf{c}'$.
Outputs 1 if $b' = b$, otherwise outputs 0.

The advantage, $\mathrm{Adv}_{\mathcal{A},\mathsf{PKE}}^{\mathsf{IND\text{-}CCA2}}(\lambda)$, of $\mathcal{A}$ in $\mathbf{Exp}_{\mathcal{A},\mathsf{PKE}}^{\mathsf{IND\text{-}CCA2}}(\lambda)$ can be defined by

$$\mathrm{Adv}_{\mathcal{A},\mathsf{PKE}}^{\mathsf{IND\text{-}CCA2}}(\lambda) = \left|2\Pr[\mathbf{Exp}_{\mathcal{A},\mathsf{PKE}}^{\mathsf{IND\text{-}CCA2}}(\lambda) = 1] - 1\right|. \tag{1}$$

A PKE scheme $\mathcal{E} = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ is IND-CCA2 secure if for any PPT $\mathcal{A}$, $\mathrm{Adv}_{\mathcal{A},\mathsf{PKE}}^{\mathsf{IND\text{-}CCA2}}(\lambda)$ is negligible with respect to λ. In addition, the above experiment can become an IND-CPA experiment if $\mathcal{A}$ is not allowed to make decryption query. We denote by $\mathbf{Exp}_{\mathcal{A},\mathsf{PKE}}^{\mathsf{IND\text{-}CPA}}(\lambda)$ the IND-CPA experiment and the advantage of $\mathcal{A}$ is defined by

$$\mathrm{Adv}_{\mathcal{A},\mathsf{PKE}}^{\mathsf{IND\text{-}CPA}}(\lambda) = \left|2\Pr[\mathbf{Exp}_{\mathcal{A},\mathsf{PKE}}^{\mathsf{IND\text{-}CPA}}(\lambda) = 1] - 1\right|. \tag{2}$$

We divide the security proof of our scheme into two phases following the approach in [20]. First, we give the IND-CPA security proof based on the IND-CPA security of the LT scheme. Then, we prove the IND-CCA2 security of our scheme in the random oracle model.

Theorem 1 *Assume that $\mathcal{A}$ is an attacker in the experiment $\mathbf{Exp}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\mathsf{PKE}}(\lambda)$ with advantage ϵ against the LT scheme. Then, we have*

$$\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\mathsf{PKE}}(\lambda) \leq \varepsilon + \frac{q_{Rd}}{q^{mr}},$$

where q_{Rd} is the query times to oracle Rd and r is the length of the output vectors of the function Rd.

Proof We assume that $\mathcal{C}$ is a challenger, who is responsible for the generation of the public parameters *params* and a public-private key pair (*pk*, *sk*) by running the algorithms Setup and KGen, respectively. It is obvious that $\mathcal{A}$ should have the advantage $\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\mathsf{PKE}}(\lambda)$ against our scheme if $\mathcal{C}$ simulates two oracles $\mathcal{H}$ and *Rd* correctly. After receiving $\mathbf{m}_0$ and $\mathbf{m}_1$ from $\mathcal{A}$, $\mathcal{C}$ randomly chooses a vector $\mathbf{v}$, a bit b and computes $\mathcal{H}$ and *Rd* as

$$Rd(\mathbf{v}) = \mathbf{u}_1 - (\mathbf{m}_b|\mathbf{con}) \quad \text{and} \quad \mathcal{H}(\mathbf{u}_1) = \mathbf{u}_2 - \mathbf{v}.$$

If $\mathcal{A}$ makes a query to *Rd* on a vector $\mathbf{v}'$, $\mathcal{C}$ first checks whether $\mathbf{v}' = \mathbf{v}$. If yes, $\mathcal{C}$ outputs $Rd(\mathbf{v}) = \mathbf{u}_1 - (\mathbf{m}_b|\mathbf{con})$. Otherwise, $\mathcal{C}$ outputs a random vector. If $\mathcal{A}$ makes queries to $\mathcal{H}$ on a vector $\mathbf{u}'_1$, $\mathcal{C}$ first checks whether $\mathbf{u}'_1 = \mathbf{u}_1$. If yes, $\mathcal{C}$ outputs $\mathcal{H}(\mathbf{u}_1) = \mathbf{u}_2 - \mathbf{v}$. Otherwise, $\mathcal{C}$ outputs a random vector. It is obvious that $\mathcal{C}$ can not simulate the *Rd* correctly if $\mathcal{C}$ doesn't know $\mathbf{u}_1$. Hence, $\mathcal{C}$ could simulate *Rd* and $\mathcal{H}$ correctly if and only if $\mathcal{A}$ made queries to $\mathcal{H}$ on $\mathbf{u}_1$ first and then to *Rd* on $\mathbf{v}$. We define two events Evt_1 and Evt_2 as follows:

- Evt_1: the event that $\mathbf{v}$ is queried to *Rd* in q_{Rd} queries before $\mathbf{u}_1$ is queried to $\mathcal{H}$.
- Evt_2: the event that $\mathbf{u}_1$ is queried to $\mathcal{H}$ in $q_{\mathcal{H}}$ times before $\mathbf{v}$ is queried to *Rd*.

It is obvious that the probability $\Pr[\mathsf{Evt}_1 \wedge \mathsf{Evt}_2] = 0$ from the definition of Evt_1 and Evt_2 and we have

$$\Pr[\mathsf{Evt}_1 \vee \mathsf{Evt}_2] = \Pr[\mathsf{Evt}_1] + \Pr[\mathsf{Evt}_2]. \tag{3}$$

While the event $\neg\mathsf{Evt}_1 \wedge \neg\mathsf{Evt}_2$ happens, $\mathcal{A}$ can not get any information about the connection between $\mathbf{m}_b|\mathbf{con}$ and $\mathbf{u}_1|\mathbf{u}_2$. Hence, the advantage of $\mathcal{A}$ is negligible in this case. For another case that $\mathsf{Evt}_1 \vee \mathsf{Evt}_2$ happens, $\mathcal{A}$ can guess b correctly. Then we have the inequation $2\Pr[\mathbf{Exp}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\mathsf{PKE}}(\lambda) = 1] \leq 2\Pr[\mathsf{Evt}_1 \vee \mathsf{Evt}_2] + 1 - \Pr[\mathsf{Evt}_1 \vee \mathsf{Evt}_2]$. That is, we have

$$\Pr[\mathbf{Exp}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\mathsf{PKE}}(\lambda) = 1] \leq \frac{1 + \Pr[\mathsf{Evt}_1 \vee \mathsf{Evt}_2]}{2}. \tag{4}$$

According to Eq. (2), we have

$$\Pr[\mathbf{Exp}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\mathsf{PKE}}(\lambda)=1]=\frac{\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\mathsf{PKE}}(\lambda)+1}{2}$$

and

$$\frac{\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\mathsf{PKE}}(\lambda)+1}{2}\leq\frac{1+\Pr[\mathsf{Evt}_1\vee\mathsf{Evt}_2]}{2},$$

i.e.,

$$\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\mathsf{PKE}}(\lambda)\leq\Pr[\mathsf{Evt}_1\vee\mathsf{Evt}_2]. \tag{5}$$

As we know, b and $\mathbf{v}$ are generated randomly by $\mathcal{C}$, so $\mathcal{A}$ could not obtain any information about them if the event $\mathsf{Evt}_1\vee\mathsf{Evt}_2$ did not happen. For instead, $\mathcal{C}$ can recover $\mathbf{u}_4$ from $\mathbf{c}$ if the event Evt_2 happens. In the other words, while event $\mathsf{Evt}_2\wedge\neg\mathsf{Evt}_1$ happens, $\mathcal{C}$ could use $\mathcal{A}$ to break the LT scheme. Thus, we have

$$\Pr[\mathsf{Evt}_2\wedge\neg\mathsf{Evt}_1]=\varepsilon. \tag{6}$$

On the other hand, the probability that $\mathbf{v}$ is exactly queried to Rd is $\frac{1}{q^{mr}}$. Hence, the probability that Evt_1 happens during the q_{Rd} queries on Rd is

$$\Pr[\mathsf{Evt}_1]\leq 1-\left(1-\frac{1}{q^{mr}}\right)^{q_{Rd}}\leq\frac{q_{Rd}}{q^{mr}}. \tag{7}$$

According to Eqs. (3), (5), (6), (7) we can get:

$$\begin{aligned}\varepsilon&=\Pr[\mathsf{Evt}_2\wedge\neg\mathsf{Evt}_1]\\&=\Pr[\mathsf{Evt}_2]\\&=\Pr[\mathsf{Evt}_2\vee\mathsf{Evt}_1]-\Pr[\mathsf{Evt}_1]\\&\geq\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\mathsf{PKE}}(\lambda)-\frac{q_{Rd}}{q^{mr}}.\end{aligned}$$

That is, we have

$$\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\mathsf{PKE}}(\lambda)\leq\varepsilon+\frac{q_{Rd}}{q^{mr}}.$$

□

Theorem 2 *Assume that $\mathcal{A}$ is an attacker in the experiment $\mathbf{Exp}^{\mathsf{IND\text{-}CCA2}}_{\mathcal{A},\mathsf{PKE}}(\lambda)$ who has the advantage ϵ against the LT scheme. Then, we have*

$$\mathrm{Adv}^{\mathsf{IND\text{-}CCA2}}_{\mathcal{A},\mathsf{PKE}}(\lambda)\leq\varepsilon+\frac{q_{Rd}}{q^{mr}}+\frac{q_{\mathcal{D}}}{q^{2mn}},$$

where q_{Rd} and $q_{\mathcal{D}}$ are the query times to the oracles Rd and $\mathcal{D}$, respectively, and r is the length of the output vectors of the function Rd.

Proof We assume that $\mathcal{C}$ is a challenger, who is responsible for the generation of the public parameters *params* and a public-private key pair (pk, sk) by running the algorithms Setup and KGen, respectively. It is clear that $\mathcal{A}$ should have the advantage $\mathrm{Adv}^{\mathsf{IND\text{-}CCA2}}_{\mathcal{A},\mathsf{PKE}}(\lambda)$ against our scheme if $\mathcal{C}$ simulates oracles $\mathcal{H}$, Rd and the decryption oracle $\mathcal{D}$ correctly. $\mathcal{C}$ simulates $\mathcal{H}$ and Rd in the same way of the proof of Theorem 1. Then, $\mathcal{C}$ uses the plaintext-extractor described in [21] to construct the decryption oracle via the following steps.

- The plaintext-extractor takes as input a ciphertext $\mathbf{c}$.
- For q_{Rd} times queries to Rd, the input and the output pairs of Rd are denoted by $(\mathbf{v}_i, V_i)$, $1 \le i \le q_{Rd}$. Similarly, the input and the output pairs of $\mathcal{H}$ are denoted by $(\mathbf{u}_{1j}, U_{1j})$, $1 \le j \le q_{\mathcal{H}}$.
- The plaintext-extractor finds the pair $(\mathbf{v}_i, V_i) = (\mathbf{u}_2 - U_{1j}, \mathbf{u}_{1j} - (\mathbf{m}|\mathbf{con}))$ s.t.

$$\mathbf{e}_0 = Conv_{t_0}(Lt_{\ell_0}(Lt_{\ell_0+\ell_1}(\mathbf{u}_{1j}|\mathbf{u}_2))), \tag{8}$$

$$\mathbf{e}_1 = Conv_{t_1}(Rt_{\ell_1}(Lt_{\ell_0+\ell_1}(\mathbf{u}_{1j}|\mathbf{u}_2)) \tag{9}$$

  are two error vectors of $\mathbf{c}$ and $Rt_k(\mathbf{u}_{1j}|\mathbf{u}_2)$ is the plaintext of $\mathbf{c}$ in the LT scheme.
- If the plaintext-extractor cannot find the pair $(\mathbf{v}_i, V_i)$ or $(\mathbf{u}_{1j}, U_{1j})$, it rejects the ciphertext $\mathbf{c}$.

Based on the above construction, we know that the plaintext-extractor cannot simulate $\mathcal{D}$ unless $\mathcal{A}$ made queries $\mathbf{u}_{1j}$ to $\mathcal{H}$ and $\mathbf{v}_i$ to Rd before the ciphertext $\mathbf{c}$ is queried to $\mathcal{D}$. It should be noted that in this case we have Eqs. (8) and (9). For further analysis, we define a new event, denoted by Evt_3, as

- Evt_3: the event that $\mathcal{A}$ queries the appropriate ciphertext $\mathbf{c}$ to $\mathcal{D}$ before $\mathcal{A}$ queries $\mathbf{u}_{1j}$ to $\mathcal{H}$ and $\mathbf{v}_i$ to Rd.

It is obvious that the probability that the appropriate ciphertext $\mathbf{c}$ is queried to $\mathcal{D}$ is $\frac{1}{q^{2mn}}$. Hence, we get the probability that the event Evt_3 occurs in at most $q_{\mathcal{D}}$ queries satisfies

$$\Pr[\mathsf{Evt}_3] \le 1 - \left(1 - \frac{1}{q^{2mn}}\right)^{q_{\mathcal{D}}} \le \frac{q_{\mathcal{D}}}{q^{2mn}}. \tag{10}$$

□

As in the proof of Theorem 1, $\mathcal{C}$ could simulate these oracles correctly while the event Evt_2 (equivalent to event $\mathsf{Evt}_2 \wedge \neg\mathsf{Evt}_3 \wedge \neg\mathsf{Evt}_1$) happened. Then, we have

$$\begin{aligned}
\varepsilon &= \Pr[\mathsf{Evt}_2 \wedge \neg\mathsf{Evt}_3 \wedge \neg\mathsf{Evt}_1] \\
&= \Pr[\mathsf{Evt}_2 \wedge \neg\mathsf{Evt}_1] - \Pr[\neg\mathsf{Evt}_2 \wedge \mathsf{Evt}_3 \wedge \mathsf{Evt}_1] \\
&\ge \Pr[\mathsf{Evt}_2 \wedge \neg\mathsf{Evt}_1] - \Pr[\mathsf{Evt}_3] \\
&\ge \Pr[\mathsf{Evt}_2] - \Pr[\mathsf{Evt}_3] \\
&\ge \Pr[\mathsf{Evt}_2 \vee \mathsf{Evt}_1] - \Pr[\mathsf{Evt}_1] - \Pr[\mathsf{Evt}_3] \\
&\ge \mathrm{Adv}^{\mathsf{IND\text{-}CCA2}}_{\mathcal{A},\mathsf{PKE}}(\lambda) - \frac{q_{Rd}}{q^{mr}} - \frac{q_{\mathcal{D}}}{q^{2mn}}.
\end{aligned}$$

That is, we have $\mathrm{Adv}_{\mathcal{A},\mathsf{PKE}}^{\mathsf{IND\text{-}CCA2}}(\lambda) \leq \varepsilon + \frac{q_{Rd}}{q^{mr}} + \frac{q_{\mathcal{D}}}{q^{2mn}}$.

Efficiency

In this section, we analyze the efficiency of our proposed PKE scheme by comparing with two recently proposed schemes (i.e., Wang's scheme [22] and Loidreau's scheme [17]), in terms of public key and ciphertext sizes. It should be noted that both Loidreau's scheme and our proposed scheme are based on rank metric codes, while Wang's scheme is based on Hamming metric codes. For a better understanding of our strengths, we introduce a family of the most representative rank metric codes called Gabidulin codes first.

Definition 7 *(Gabidulin codes)* For an element $a \in \mathbb{F}_{q^m}$ and an integer $i \in \mathbb{Z}$, we denote $a^{[i]} = a^{q^i}$. Further for a vector $\mathbf{a} = (a_0, a_1, \ldots, a_{n-1}) \in \mathbb{F}_{q^m}^n$, we denote $\mathbf{a}^{[i]} = (a_0^{[i]}, a_1^{[i]}, \ldots, a_{n-1}^{[i]})$. Then, an $[n,k]_{\mathbb{F}_{q^m}}$ Gabidulin code $\mathcal{C}(\mathbf{a})$ for a vector $\mathbf{a} = (a_0, a_1, \ldots, a_{n-1}) \in \mathbb{F}_{q^m}^n$ is defined with the following k by n generator matrix:

$$\begin{pmatrix} a_0^{[0]} & a_1^{[0]} & \cdots & a_{n-1}^{[0]} \\ a_0^{[1]} & a_1^{[1]} & \cdots & a_{n-1}^{[1]} \\ \vdots & \vdots & \ddots & \vdots \\ a_0^{[k-1]} & a_1^{[k-1]} & \cdots & a_{n-1}^{[k-1]} \end{pmatrix}.$$

According to the analysis in [18, 23–25], a secure code-based PKE scheme should be able to resist several structural attacks, such as key recovery attack, reduction attack and moore decomposition attack [23]. In particular, we can obtain the lower bound of the computation costs that an adversary needs to break a PKE scheme based on Gabidulin codes. The lower bound, known as the PQ-security level [17, 26], can be computed by the following formula:

$$m^3 \cdot 2^{(t-1)\lfloor \frac{k \cdot \min\{m,n\}}{2n} \rfloor}, \tag{11}$$

where t is the correction ability of the underlying Gabidulin codes.

As the same in the LT scheme [18], we choose two sets of parameters $params_1 = (q = 2, m = 75, n = 73, k = 21, t_0 = t_1 = 13)$ and $params_2 = (q = 2, m = 85, n = 83, k = 18, t_0 = t_1 = 16)$, which can achieve the security levels $2^{141} \approx 75^3 \cdot 2^{\cdot(13-1)\lfloor \frac{21 \cdot \min\{75,73\}}{146} \rfloor}$ and $2^{154} \approx 85^3 \cdot 2^{\cdot(16-1)\lfloor \frac{18 \cdot \min\{85,83\}}{166} \rfloor}$, respectively, according to Eq. (11). Recall that the ciphertext in our scheme is of the form $\mathbf{c} = (\mathbf{c}_0, \mathbf{c}_1)$, where $\mathbf{c}_0$ and $\mathbf{c}_1$ are two vectors of the same length n over $\mathbb{F}_{q^m}$. For the first parameter set $params_1$, we have the ciphertext size of $2 \cdot 73 \cdot 75$ bits $= 10950$ bits ≈ 1.37 KB. Similarly, we can get the ciphertext size of $2 \cdot 83 \cdot 85$ bits $= 14110$ bits ≈ 1.76 KB for the second parameter set $params_2$.

As to the public key, it has the form of $pk = (\hat{G}, \mathbf{x})$ in our scheme, where $\hat{G}$ is a $k \times n$ matrix over $\mathbb{F}_{q^m}$ and $\mathbf{x}$ is a vector of the length n over $\mathbb{F}_{q^m}$. For the first parameter set $params_1$, we have the public key size of $75 \cdot (21 + 1) \cdot 73$ bits $= 120450$ bits ≈ 15.06 KB. Similarly, we can get the public key

size of $85 \cdot (18+1) \cdot 83$ bits $= 134045$ bits ≈ 16.76 KB for the second parameter set $param_2$.

We note that the ciphertext and public key sizes of our IND-CCA2 secure scheme are exactly the same to the ones of the IND-CPA secure LT scheme. Compared to Loidreau's scheme [17], which is a recently proposed PKE scheme based on rank metric codes, our PKE scheme has a significant advantage in terms of public key size and ciphertext size (see Table 2).

For a fair comparison between our rank metric codes-based PKE scheme and Wang's scheme from Hamming metric codes [22], we choose two new sets of parameters $params_3 = (q = 2, m = 71, n = 69, k = 19, t_0 = t_1 = 13)$ and $params_4 = (q = 2, m = 101, n = 99, k = 22, t_0 = t_1 = 17)$, which can achieve the security levels $71^3 \cdot 2^{\cdot(13-1)\lfloor \frac{19 \cdot \min\{71,69\}}{138} \rfloor} \approx 2^{133}$ and $101^3 \cdot 2^{\cdot(17-1)\lfloor \frac{22 \cdot \min\{101,99\}}{198} \rfloor} \approx 2^{196}$, respectively, according to Eq. (11). For the parameter set $params_3$, we have the ciphertext size of $2 \cdot 69 \cdot 71$ bits $= 9798$ bits ≈ 1.22 KB and public key size of $71 \cdot (19+1) \cdot 69$ bits $= 97890$ bits ≈ 12.25 KB. Similarly, we can get the ciphertext size of $2 \cdot 99 \cdot 101$ bits $= 19998$ bits ≈ 2.50 KB and public key size of $101 \cdot (22+1) \cdot 99$ bits $= 229977$ bits ≈ 28.75 KB for the parameter set $params_4$. From Table 2, we observe that our code-based PKE scheme also has a significant advantage to Wang's scheme in terms of public key size.

A secure cloud storage use case

In this section, we explain how our proposed code-based PKE scheme can be deployed in a cloud storage system. We use the notations $\mathsf{CBEnc}(\cdot)/\mathsf{CBDec}(\cdot)$ and $\mathsf{SEnc}(\cdot)/\mathsf{SDec}(\cdot)$ to denote the encryption/decryption algorithms in our proposed code-based PKE scheme and a secure symmetric encryption scheme, respectively. In our cloud storage system, we assume that there are two users (Alice and Bob) and multiple storage servers. Alice splits her data into some data blocks and sends each block to storage servers (one data block can be sent to several storage severs). Bob can get the data blocks from the cloud and recover them into the original data (see Fig. 5).

If Alice wants to store data M in the cloud and shares it with Bob, she can select a random session key K for the symmetric encryption algorithm and encrypts it under Bob's public key pk_B to get a ciphertext $EK = \mathsf{CBEnc}(pk_B, K)$. Then, Alice can split

Table 2 Performance evaluation of our scheme, Loidreau's scheme [17] and Wang's scheme [22]: a comparative summary

	q^m	n	k	t	Ciphertext size (KB)	Public key size (KB)	Security level	Semantic security
Loidreau's scheme	2^{128}	90	24	11	1.44	21.50	140	IND-CCA2
	2^{128}	120	80	4	1.92	51.00	141	
Wang's scheme	2^9	1020	660	180	1.15	980	128	Not Given
	2^{10}	1560	954	203	1.95	2460	192	
Our PKE scheme	2^{75}	73	21	13	1.37	15.06	141	IND-CCA2
	2^{85}	83	18	16	1.76	16.76	154	
	2^{71}	69	19	13	1.22	12.25	133	
	2^{101}	99	22	17	2.5	28.75	196	

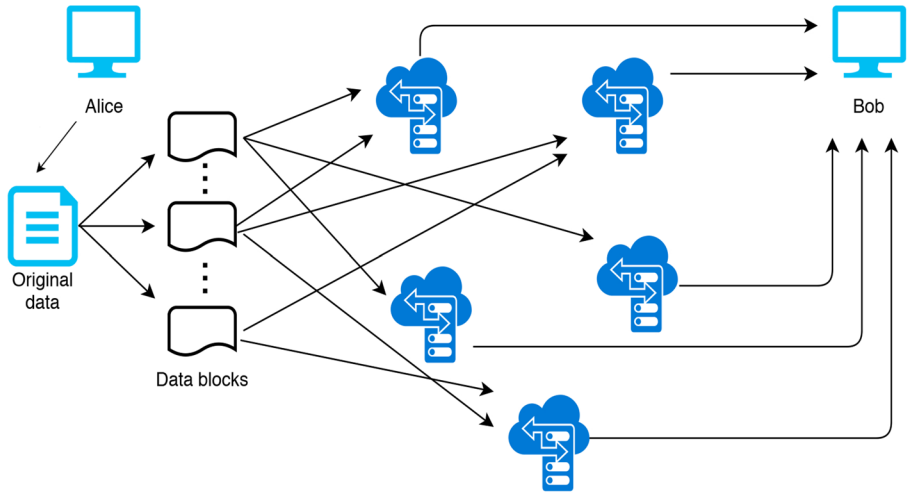

Fig. 5 High level cloud storage architecture

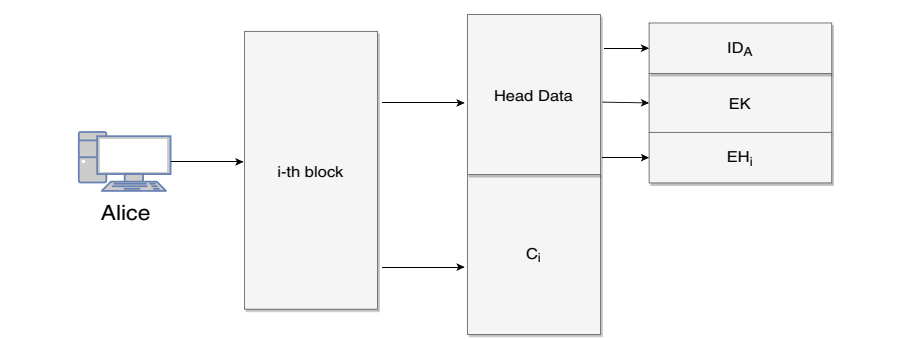

Fig. 6 Structure of uploaded data

the data M into m blocks $M = M_1 \| M_2 \| \cdots \| M_m$ and uses K to encrypt them to get $C_i = \mathsf{SEnc}(K, M_i)$, $1 \le i \le m$. Next, Alice computes the hash values $h_i = Hash(C_i)$, $1 \le i \le m$, and saves them in her local storage, where $Hash(\cdot)$ is a secure collision-resistant hash function. Then, Alice sets the information ID_A, EK, and $EH_i = \mathsf{SEnc}(K, h_i)$ to each ciphertext block C_i, $1 \le i \le m$, as the header data (see Fig. 6). Finally, she sends the header data and ciphertext C_i, $1 \le i \le m$, to m storage servers. If Alice wants to check the data integrity of some ciphertext block C_i, she can ask the corresponding storage server to compute $Hash(C_i)$ and send it back to her. If the received $Hash(C_i)$ matches her local hash value for C_i, she is assured that the data are intact and have not modified by the cloud storage server.

On the other hand, if Bob wants to get the data M from Alice via the cloud storage system, he can find the corresponding m ciphertext data blocks C_i, $1 \le i \le m$, and their header data (ID_A, EK, EH_i) by searching for ID_A in the cloud. Then, Bob can download them and runs $\mathsf{CBDec}(sk_B, EK)$ to recover the symmetric key K with his private key sk_B. Next, for each $1 \le i \le m$, Bob uses the obtained symmetric key K to decrypt EH_i and C_i to get $Hash(C_i)$ and M_i, respectively. To verify the data integrity of

M_i, Bob computes $Hash(C_i)$ and checks whether it is equal to h_i obtained from EH_i. If yes, it means that C_i has not been modified by the storage server and the corresponding M_i is the original data from Alice. Otherwise, Bob asks the corresponding SS_i of the non-matching C_i to send the data again.

From the above description, we find that the security of the proposed cloud storage system relies mainly on the security of our code-based PKE scheme and the symmetric encryption scheme. As mentioned in "Introduction" section, our scheme is able to resist quantum computer-facilitated attacks and achieves IND-CCA2 security as demonstrated in "Security proof" section. Clearly, Alice can access her data stored in the cloud using any devices by searching for the data with her ID. Moreover, both Alice and Bob can check the integrity of the downloaded data from the cloud. In the event that the data were modified by the cloud storage server or during the communication, Bob can detect this and then proceed to download the corresponding blocks again. Thus, the proposed cloud storage system provides availability, reliability, efficient retrieval and data sharing, even in the post-quantum era.

Conclusion

Code-based public key cryptography (PKC) can potentially be deployed in real-world applications to ensure quantum computing attack resilience. Existing code-based PKC schemes are broadly categorized into those based on Hamming metric codes and those based on rank metric codes, and it is believed that the latter generally has smaller public key and ciphertext sizes than the former under the same security level.

In this paper, we presented a new rank metric codes-based public key encryption scheme from Lau and Tan's scheme [18], and hence inherits the latter's small public key and ciphertext size properties. However, our new scheme achieves IND-CCA2 secure (the highest security assurance possible), as shown in this paper. We also analyzed its efficiency in term of the public key size and the ciphertext size. Finally, we presented a use case to demonstrate its utility in practice.

Future work includes constructing more rank metric codes-based cryptographic primitives, such as proxy re-encryption and attribute-based encryption, to achieve other desirable properties in practical applications. We also intend to implement and evaluate a prototype of the proposed/extended scheme in collaboration with a (small) cloud storage service provider.

Acknowledgements
We would like to thank the reviewers for their valuable comments.

Authors' contributions
The authors have contributed significantly to the research work presented of this manuscript. All authors read and approved the final manuscript.

Funding
The work is supported in part by the National Key R&D Program of China under Grant No. 2017YFB0802300, the NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informatization under Grant No. U1509219, the Shanghai Natural Science Foundation under Grant No. 17ZR1408400, the National Natural Science Foundation of China under Grant Nos. 61601129, 11701179, the Shanghai Science and Technology Commission Program under Grant No. 18511105700, and the Shanghai Sailing Program under Grant No. 17YF1404300.

Availability of data and materials
The datasets used and analysed in this study are available from the corresponding author on reasonable request.

Competing interests
The authors declare that they have no competing interests.

Author details
[1] Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai, China. [2] College of Computer Science and Technology, Shanghai University of Electric Power, Shanghai, China. [3] Department of Information Systems and Cyber Security, The University of Texas at San Antonio, San Antonio, TX 78249, USA.

Received: 31 December 2018 Accepted: 16 August 2019
Published online: 02 September 2019

References

1. Rathore S, Sharma PK, Park JH (2017) XSSClassifier: an efficient XSS attack detection approach based on machine learning classifier on SNSs. J Inf Process Syst 13(4):1014–1028
2. Rathore S, Park JH (2018) Semi-supervised learning based distributed attack detection framework for IoT. Appl Soft Comput 72:79–89
3. Rathore S, Loia V, Park JH (2018) SpamSpotter: an efficient spammer detection framework based on intelligent decision support system on Facebook. Appl Soft Comput 67:920–932
4. Rathore S, Sangaiah AK, Park JH (2018) A novel framework for internet of knowledge protection in social networking services. J Comput Sci 26:55–65
5. Kamara S, Lauter K (2010) Cryptographic cloud storage. In International conference on financial cryptography and data security. Springer, Berlin, pp 136–149
6. Liu Q, Wang G, Wu J (2014) Time-based proxy re-encryption scheme for secure data sharing in a cloud environment. Inf Sci 258:355–370
7. Mishra B, Jena D (2018) CCA secure proxy re-encryption scheme for secure sharing of files through cloud storage. In 2018 fifth international conference on emerging applications of information technology (EAIT). IEEE, Piscataway, pp 1–6
8. Li J, Zhang Y, Chen X, Xiang Y (2018) Secure attribute-based data sharing for resource-limited users in cloud computing. Comput & Secur 72:1–12
9. Liu Z, Li T, Li P, Jia C, Li J (2018) Verifiable searchable encryption with aggregate keys for data sharing system. Future Gener Comput Syst 78:778–788
10. Chu CK, Chow SSM, Tzeng WG, Zhou J, Deng RH (2014) Key-aggregate cryptosystem for scalable data sharing in cloud storage. IEEE Trans Parallel Distrib Syst 25(2):468–477
11. Haeger ES, Schurig K, Cenname M, et al. (2015) Crypto proxy for cloud storage services: U.S. Patent 9,137,222[P]. 2015-9-15
12. Xu L, Wu X, Zhang X (2012) CL-PRE: a certificateless proxy re-encryption scheme for secure data sharing with public cloud. In: Proceedings of the 7th ACM symposium on information, computer and communications security. ACM, New York, pp 87–88
13. Shor PW (1994) Algorithms for quantum computation: Discrete logarithms and factoring. Proceeding of FOCS 1994, Santa Fe, New Mexico, November 20–22, 1994, pp 124-134
14. Shor PW (1999) Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Rev 41(2):303–332
15. Mceliece RJ (1978) A public-key cryptosystem based on algebraic coding theory. Deep Space Netw Prog Rep 42–44(1978):114–116
16. Bernstein DJ (2009) Introduction to post-quantum cryptography. In: Post-quantum cryptography, Springer, Berlin. pp 1–14
17. Loidreau P (2017) A new rank metric codes based encryption scheme. In: Lange T, Takagi T (eds.) PQCrypto 2017. LNCS, 10346, pp 3–17
18. Lau TSC, Tan CH (2018) A new encryption scheme based on rank metric codes. In: Australasian conference on information security and privacy. pp 750–758
19. Gaborit P, Hauteville A, Phan DH, Tillich J-P (2017) Identity-based encryption from codes with rank metric. In: Katz J, Shacham H (eds.) CRYPTO 2017. LNCS, 10403, pp 194–224
20. Kobara K, Imai H (2001) Semantically secure McEliece public-key cryptosystems-conversions for McEliece PKC. Int Workshop Pract Theory Public Key Cryptogr Public Key Cryptogr 1992:19–35
21. Bellare M, Rogaway P (1994) Optimal asymmetric encryption. Eurocrypto 950(6):92–111
22. Wang Y (2016) Quantum resistant random linear code based public key encryption scheme RLCE. IEEE Int Symp Inf Theory (ISIT) 2016:2519–2523
23. Horlemann-Trautmann AL, Marshall K, Rosenthal J (2015) Extension of overbeck's attack for gabidulin based cryptosystems. Des Codes Cryptogr 86(2):1–22
24. Otmani A, Kalachi HT, Ndjeya S (2018) Improved cryptanalysis of rank metric schemes based on Gabidulin codes. Des Codes Cryptogr 86(9):1983–1996
25. Gaborit P, Zémor G (2016) On the hardness of the decoding and the minimum distance problems for rank codes. IEEE Trans Inf Theory 62(12):7245–7252
26. Bernstein DJ (2010) Grover vs. mceliece. Post-Quantum Cryptography 2010. Lecture Notes Comput Sci 6061:73–80

Publisher's Note