Annales Mathematicae et Informaticae
Accepted manuscript
DOI: 10.33039/ami.2026.01.001
URL: https://ami.uni-eszterhazy.hu

Symmetric and asymmetric encryption schemes based on ill-posed inverse problems

Gaurav Mittal, Saibal Kumar Pal

Defence Research and Development Organization,
Near Metcalfe House, New Delhi, 110054, India
gaurav.mittaltwins@gmail.com
saibal.pal@gov.in

Abstract. In this work, we unveil an analogy between the well-known lattice based learning with error problem and ill-posed inverse problems. We show that LWE problem is a structured inverse problem. Further, we propose a symmetric encryption scheme based on ill-posed problems and thoroughly discuss its security. Finally, we propose a public key encryption scheme based on our symmetric encryption scheme and CRYSTALS-Kyber KEM (key encapsulation mechanism) and discuss its security.

Keywords: learning with error problem, cryptography, inverse problems

2020 *Mathematics Subject Classification:* Primary 94A60, 65J22, 20C07

1. Introduction and analogy

1.1. Learning With Error problem

The LWE (Learning With Error) problem is a well-known hard problem in cryptography [3]. After undergoing nearly a decade of rigorous scrutiny, the crypto-primitives CRYSTALS Kyber [5] and CRYSTALS Dilithium [7], whose security inherently rely on LWE problem, have been standardized by NIST. The LWE problem was put forward by Regev in his seminal work [20]. Formally, one may define the LWE problem as follows:

Definition 1.1. For a prime q, let $\mathcal{D}$ be an error distribution over the modular ring $\mathbb{Z}_q$. For a given m (also known as dimension parameter), let $s \leftarrow \mathbb{Z}_q^m$ be chosen

Submitted: August 31, 2025
Accepted: January 7, 2026
Published online: January 8, 2026

uniformly at random and it is kept as a secret. For $1 \leq j \leq n$, where n is some polynomial in m, consider the samples

$$(a_j, b_j = \langle a_j, s \rangle + e_j) \in \mathbb{Z}_q^m \times \mathbb{Z}_q,$$

where $a_j \leftarrow \mathbb{Z}_q^n$ is picked uniformly at random. Further, $e_j \leftarrow \mathbb{Z}_q$ is picked following distribution $\mathcal{D}$ and typically, this is a short element (in terms of norm). The LWE problem is to derive s from the knowledge of $(a_j, b_j)_{1 \leq j \leq n}$.

The LWE problem can also be formulated in terms of matrix form as follows: Given

$$\mathbf{A} = \begin{bmatrix} (a_1) \\ (a_2) \\ \vdots \\ (a_n) \end{bmatrix}_{m \times n}, \quad \mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix} \in \mathbb{Z}_q^n, \quad \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} = \begin{bmatrix} (a_1) \\ (a_2) \\ \vdots \\ (a_n) \end{bmatrix} \times \mathbf{s} + \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix} \in \mathbb{Z}_q^n,$$

we need to find $\mathbf{s} \in \mathbb{Z}_q^m$ from the knowledge of $\mathbf{A}, \mathbf{b}$.

It is well-known that the LWE problem is very hard and solving it quantumly (using quantum computers), on an average, is equivalent to solving certain lattice problems in the worst-case [20]. This makes LWE a very strong candidate for post-quantum cryptography and the worst-case advantage is not associated with any other post-quantum candidates [3]. Next, to explore the roots of computational hardness of LWE problem, we switch into the subject of inverse problems. We refer to [8] for more preliminary details on this subject.

1.2. Well-posed and ill-posed problems

We consider the operator equations of the form

$$T \colon D(T) \subset U \to V \quad \text{defined as} \quad T(u) = v, \tag{1.1}$$

where U and V denote Banach spaces and $D(T)$ denotes the domain of T. Given u, deducing $v = T(u)$ is the direct problem. The corresponding inverse problem is deducing u from the value of v. As per Hadamard criteria, a problem of the form (1.1) is well-posed if the following three conditions are met.

(i) For a given $v \in V$, (1.1) has a solution.

(ii) The solution is unique.

(iii) There must be continuous dependency, i.e., if v' is given in place of v and it is close to v, then the corresponding solutions u' and u should be near to each other (in terms of norm).

Further, if any one of the above three conditions is not satisfied, then the problem (1.1) is ill-posed. It is worth to note that the conditions (i)-(ii) are straight-forward to understand. The condition (iii) is less clear. Before going further, we discuss example of an ill-posed problem that does not fulfill condition (iii).

Example of an ill-posed problem

Let $L^2[0,1]$ be the space of real-valued Lebesgue integrable functions g defined on $[0,1]$ such that $\int_0^1 g^2(y)\,dy < \infty$. We consider an operator

$$\mathcal{S} : L^2[0,1] \to L^2[0,1]$$

defined as

$$(\mathcal{S}(\Psi))(y) = \int\limits_0^1 e^{-|y-s|}\Psi(s)\,\mathrm{d}s. \tag{1.2}$$

This operator is also known as Hilbert–Schmidt operator (HSO). For the integral equation (1.2), the kernel $\mathcal{K}(y,s) = e^{-|y-s|}$ is square-integrable as well as continuous. Therefore, HSO is a compact operator. Further, we know that any compact operator possesses a singular value decomposition (see [8]). Consequently, we may write

$$\mathcal{S}(\Psi) = \sum_{k=1}^{\infty} s_k \langle \Psi, \beta_k \rangle \alpha_k,$$

where $\{s_k\}$ are singular values of $\mathcal{S}$ satisfying $s_k \to 0$ as $k \to \infty$ and $\{\alpha_k\}, \{\beta_k\}$ are orthonormal systems. For the operator $\mathcal{S}$, its inverse operator can be written as

$$\Psi = \mathcal{S}^{-1}(\Phi) = \sum_{k=1}^{\infty} \frac{1}{s_k} \langle \Phi, \alpha_k \rangle \beta_k. \tag{1.3}$$

We assume the availability of perturbed data since exact data is not available in practice. So, we may write

$$\Phi = \mathcal{S}(\Psi) + \mathcal{E}.$$

Here $\mathcal{E}$ represents the small error. Applying $\mathcal{S}^{-1}$ on this Φ using (1.3) to derive that

$$\mathcal{S}^{-1}(\Phi) = \Psi + \mathcal{S}^{-1}(\mathcal{E}) = \Psi + \sum_{k=1}^{\infty} \frac{1}{s_k} \langle \mathcal{E}, \alpha_k \rangle \beta_k. \tag{1.4}$$

Since the operator $\mathcal{S}$ is compact, we note that

$$s_k \to 0 \quad \text{as} \quad k \to \infty \implies \frac{1}{s_k} \to \infty \quad \text{as} \quad k \to \infty.$$

This and (1.4) derive that

$$\|\mathcal{S}^{-1}(\Phi) - \Psi\|_{L^2[0,1]} > C$$

for any positive constant $C > 0$. Therefore, there is no continuous dependence between the data and the solution. Hence, the operator $\mathcal{S}$ is ill-posed in the sense of Hadamard due to violation of condition (iii).

1.3. Degree of ill-posedness

In this subsection, we discuss about the operator equations of the form (1.1) with the additional constraint that T is a compact operator between Hilbert spaces. The singular value decomposition (SVD) theorem implies that T has singular values

$$s_1 \geq s_2 \geq s_3 \geq \cdots \geq s_k > 0.$$

We note that (1.1) is ill-posed if

$$s_k \to 0 \quad \text{as} \quad k \to \infty.$$

This is also evident from (1.4) as decaying values of s_k leads to instability of A^{-1}. Further, the degree of ill-posedness can be described by looking at rate at which singular value decays. Accordingly, we have the following two categories of ill-posed problems.

(i) **Mildly ill-posed:** If the decay is of the form

$$s_k \sim k^{-t}, \quad t > 0,$$

i.e., s_k decays polynomially (or algebraic decay), then (1.1) is mildly ill-posed.

(ii) **Severely ill-posed:** If the decay is of the form

$$s_k \sim e^{-rk}, \quad r > 0,$$

i.e., s_k decays exponentially, then (1.1) is severely ill-posed.

We note that faster the rate at which s_k's decay, more the degree of ill-posedness of (1.1).

1.4. Structural analogy

In this subsection, we discuss the analogy between LWE problem and ill-posed inverse problems. For LWE problem, the solution $\mathbf{s}$ belongs to finite dimensional space $\mathbb{Z}_q^m$. However, for the inverse problem associated with HSO, the solution Ψ belongs to the infinite dimensional space $L^2[0,1]$. Further, for both LWE problem and inverse problem for HSO, the common part is the availability of noisy samples or data. This is the main reason that lead to difficulty in determining their respective solutions. For LWE problem, presence of noisy samples makes it computationally infeasible to determine the solution $\mathbf{s}$ whereas for inverse problem related to HSO, noisy data leads to its ill-posedness and therefore, additional regularization techniques are needed to find the approximate solution. The most important point of difference is the type of noise present in both the problems. For LWE problem, noise is discrete and it is added deliberately so that determination of $\mathbf{s}$ becomes computationally hard whereas for inverse problem for HSO, noise is analytic and continuous in nature and it is always present in real-world measurements. This whole discussion is also given in brief in the following Table 1.

Table 1. Analogy between LWE problem and Inverse problems.

Parameters	**LWE Problem**	**Inverse Problem for HSO**
Dimension	Finite: modular matrix multiplication	Infinite: Operator $\mathcal{S} : L^2[0,1] \to L^2[0,1]$
Data	$\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$ with noise $\mathbf{e}$ in finite dimensional space	$\Phi = \mathcal{S}(\Psi) + \mathcal{E}$ with noise $\mathcal{E}$ in infinite dimensional space
Solution	Deduce $\mathbf{s}$ from noisy samples $\mathbf{b}$ but it is computationally hard due to presence of noise $\mathbf{e}$	Deduce Ψ from noisy data Φ. Noise presence makes it highly unstable and therefore regularization techniques are needed to reconstruct the exact solution
Noise	Discrete in nature and makes deduction of secret $\mathbf{s}$ computationally infeasible Here, noise is added deliberately	Analytic and continuous in nature and, in practice, noisy data is available in place of exact data

We note the following:

- Both LWE problem and Inverse problem for HSO demand to invert a linear operator, which is contaminated by noise. As a result, inversion is either unstable or computationally hard.
- In inverse problem for HSO, a small noise $\mathcal{E}$ makes the problem highly ill-posed. In LWE problem, noise plays the role of a mask, which makes it computationally infeasible to deduce the secret.

In this manner, we can see that LWE problem is a structured inverse problem. Precisely, LWE problem can be seen as a special case of inverse problems, i.e.,

$$\text{LWE problem} \in \{\text{P} : \text{P is an ill-posed inverse problem}\}.$$

In inverse problems, we aim for approximate solution (which is close to exact solution in terms of norm) using regularization techniques whereas for LWE problem, we look for the exact solution. If noise samples in inverse problems are discrete in nature, then LWE problem and inverse problem with matrix operator are same.

2. Symmetric encryption scheme based on ill-posed problems

In this section, we propose a symmetric encryption scheme based on ill-posed inverse problems. This is defined through the following subsections.

2.1. Parameters and key generation

- Let $\mathcal{S} : L^2[0,1] \to L^2[0,1]$ be a compact operator. Let $\mathcal{S}^{-1}$ be the inverse of $\mathcal{S}$. We remark that both $\mathcal{S}$ and $\mathcal{S}^{-1}$ are known publically.

- Let $\mathcal{E}$ be the error and let it follow certain distribution, e.g., discrete Gaussian distribution or centered binomial distribution. This error is chosen secretly and acts as a secret key. Further, the error space has atleast $N \geq 2^{128}$ elements. For example, we take $N = 2^{128}$ and partition $[0, 1]$ into subintervals

  $$\mathcal{I}_j = \left[\frac{j-1}{N}, \frac{j}{N}\right), \quad 1 \leq j \leq N.$$

  We note that the subintervals I_j's are disjoint. We fix $\varepsilon > 0$ and define

  $$h_j(x) := \sqrt{N\varepsilon^2}\, \mathbf{1}_{I_j}(x), \quad 1 \leq j \leq N.$$

  Here, $\mathbf{1}_{I_j}$ represents the indicator function of I_j. It can be easily noted that $h_j \in L^2[0, 1]$ for each j and norm of each h_j is ε as

  $$\|h_k\|_2 = \sqrt{N\varepsilon^2}\, \|\mathbf{1}_{I_j}\|_2 = \sqrt{N\varepsilon^2} \cdot \sqrt{\frac{1}{N}} = \varepsilon$$

  since $\|\mathbf{1}_{I_j}\|_2 = \frac{1}{\sqrt{N}}$. We consider the error set as $\{h_1, h_2, \ldots, h_N\}$. Clearly, this set has cardinality $N = 2^{128}$. Further, all the errors are of small size ϵ.

2.2. Encoding

- Let $\{0, 1\}^t$ be the message space containing all bit-strings of length t.
- Let $\mu \in \{0, 1\}^t$ be an arbitrary message string. Then one may see the space $\{0, 1\}^t$ as a subset of $L^2[0, 1]$ using one of the following two mappings.
- **Map-1**: Using Fourier or Haar basis, generate an orthonormal sequence $\{e_k\} \subset L^2[0, 1]$. We enumerate elements in $\{0, 1\}^t$ as $\{\mu_1, \mu_2, \ldots, \mu_{2^t}\}$. Then consider the map

  $$\wp_1 \colon \{0, 1\}^t \to L^2[0, 1] \quad \text{defined as} \quad \wp_1(\mu_k) = e_k. \tag{2.1}$$

  This map is injective.
- **Map-2**: This map is defined using piecewise constant functions. Consider a string $\mu = s_1 s_2 \cdots s_t \in \{0, 1\}^t$. We define $\Psi_\mu \in L^2[0, 1]$ as

  $$\Psi_\mu(y) = s_i, \quad \text{if } y \in \left[\frac{j-1}{t}, \frac{j}{t}\right),\ 1 \leq j \leq t.$$

  It can be verified that $\Psi_\mu \in L^2[0, 1]$. Accordingly, we define the map

  $$\wp_2 \colon \{0, 1\}^t \to L^2[0, 1] \quad \text{defined as} \quad \wp_2(\mu) = \Psi_\mu. \tag{2.2}$$

  This map is also injective.
- Clearly, $\wp_1$, $\wp_2$ represent the desired encoding of binary messages as elements of $L^2[0, 1]$.

2.3. Encryption

- Let $\mu = s_1 s_2 \cdots s_t \in \{0,1\}^t$ be the message to be encrypted.
- Apply $\wp_1$ given by (2.1) (or $\wp_2$ given by (2.2)) on μ to obtain $\wp_1(\mu) \in L^2[0,1]$.
- The ciphertext is

  $$\mathcal{C} = \mathcal{S}(\wp_1(\mu)) + \mathcal{E}, \tag{2.3}$$

  where $\mathcal{E}$ is the secret key. This error introduces noise in the data.

2.4. Decryption and decoding

- The decryptor after receiving $\mathcal{C}$, uses secret key $\mathcal{E}$ to obtain

  $$\mathcal{C} - \mathcal{E} = \mathcal{S}(\wp_1(\mu)).$$

- Apply $\mathcal{S}^{-1}$ on the exact data to obtain

  $$\mathcal{S}^{-1}(\mathcal{C} - \mathcal{E}) = \wp_1(\mu).$$

- Represent $\wp_1(\mu)$ as an element of $\{0,1\}^t$ by using $\wp_1^{-1}$ (which exists on its range). This yields the message.

Next, we discuss the security of our encryption scheme along with its certain characteristics.

2.5. Security analysis

2.5.1. Brute force attack

The adversary needs to try all the possible errors $\mathcal{E}$ from the set of errors to get the message from the knowledge of $\mathcal{C}$. In our case, the error space has N elements, where N is atleast 2^{128} (see subsection 2.1). A brute-force search over this space is not computationally feasible, as no polynomial-time algorithm can exhaustively explore a set of this size, even under the most powerful currently conceivable computational resources.

2.5.2. Error should be ephemeral

Suppose two messages μ_1 and μ_2 are such that same error $\mathcal{E}$ is taken to encrypt both. Then, (2.3) implies that

$$\begin{aligned} \mathcal{C}_1 &= \mathcal{S}(\wp_1(\mu_1)) + \mathcal{E} \\ \mathcal{C}_2 &= \mathcal{S}(\wp_1(\mu_2)) + \mathcal{E}. \end{aligned}$$

These two imply that

$$\mathcal{C}_1 - \mathcal{C}_2 = \mathcal{S}(\wp_1(\mu_1)) - \mathcal{S}(\wp_1(\mu_2)).$$

If the operator $\mathcal{S}$ is linear, then this means

$$\mathcal{C}_1 - \mathcal{C}_2 = \mathcal{S}(\wp_1(\mu_1) - \wp_1(\mu_2)). \tag{2.4}$$

Further, if $\wp_1$ is also linear, then (2.4) gives the encryption of $\mu_1 - \mu_2$. Therefore, for every message, error $\mathcal{E}$ should be chosen uniformly at random. In our case, the probability that $\mathcal{E}$ is same in $\mathcal{C}_1$ and $\mathcal{C}_2$ is 2^{-128} since the error space contains 2^{128} elements and error is chosen uniformly at random for each message. This probability is very near to 0. Additionally, one can further reduce this probability by increasing N.

2.5.3. Probabilistic encryption

Our scheme comes under the category of probabilistic encryption [10]. Given a fixed message μ, there can be many ciphertexts corresponding to this message. This is because by changing the error, ciphertext gets changed corresponding to a fixed message.

2.5.4. Security against ciphertext only attack

The adversary $\mathfrak{A}$ knows $\mathcal{C}$ and $\mathcal{S}^{-1}$. Using these in (2.3), $\mathfrak{A}$ computes

$$\mathcal{S}^{-1}(\mathcal{C}) = \wp_1(\mu) + \mathcal{S}^{-1}(\mathcal{E}).$$

But due to ill-posedness of the compact operator equation $\mathcal{S}(\Psi) = \Phi$ (or inverse problem), the small error $\mathcal{E}$ leads to large error in $\mathcal{S}^{-1}(\mathcal{C})$ as shown in subsection 1.2. Therefore, it becomes computationally infeasible for $\mathfrak{A}$ to deduce μ without applying regularization techniques. We remark that the application of regularization techniques further depends on the degree of ill-posedness of the inverse problem as discussed in subsection 1.3.

2.5.5. Security against known/chosen plaintext attacks

The adversary knows (or demands) polynomially many plaintext-ciphertext pairs $(\mu_k, \mathcal{C}_k)$ (these may be chosen adaptively). The main purpose of $\mathfrak{A}$ is to derive $\mathcal{E}$. It follows from (2.3) that

$$\mathcal{C}_k = \mathcal{S}(\wp_1(\mu_k)) + \mathcal{E}_k, \quad k \geq 1.$$

Since $\mathcal{E}_k$ is different and randomly chosen for every message, it becomes computationally infeasible to deduce $\mathcal{E}$ for a new message.

2.5.6. CCA2 security (security against adaptive chosen ciphertext attack)

Due to inclusion of unique/random errors at each instance, it would not be possible for $\mathfrak{A}$ to deduce the key/encryption of new message from previous adaptive plaintext-ciphertext queries in polynomial time. Thus, the scheme is CCA2 secure.

2.5.7. Security against differential cryptanalysis

We recall that for a block cipher, differential cryptanalysis looks for differential trails [12]. Let E_k denote an encryption scheme with key k and let μ, μ' denote two different plaintexts. Then differential cryptanalysis studies

$$\Delta\mu = \mu \oplus \mu' \implies \Delta\mathcal{C} = E_k(\mu) \oplus E_k(\mu').$$

In our scheme, we have

$$\mathcal{C} = \mathcal{S}(\wp_1(\mu)) + \mathcal{E}, \quad \mathcal{C}' = \mathcal{S}(\wp_1(\mu')) + \mathcal{E}'.$$

Taking their difference yields

$$\Delta\mathcal{C} = \mathcal{C} - \mathcal{C}' = (\mathcal{S}(\wp_1(\mu)) - \mathcal{S}(\wp_1(\mu')) + (\mathcal{E} - \mathcal{E}').$$

By definition of $\mathcal{S}$, we can write above as

$$\mathcal{C} - \mathcal{C}' = (\mathcal{S}(\wp_1(\mu) - \wp_1(\mu')) + (\mathcal{E} - \mathcal{E}').$$

Any stable differential propagation in our case would be destroyed by the noise term $\mathcal{E} - \mathcal{E}'$. Consequently, differential cryptanalysis does not provide any advantage.

2.5.8. Quantum security

Unlike classical schemes such as RSA and elliptic curve cryptography [17], our scheme relies on analytical inverse problems. Therefore, Shor's algorithm [16] has no impact on our scheme. Additionally, Grover's searching algorithm [9] offers no quantum speedup. This is because the structure of $L^2[0,1]$ is different from linear algebra over finite fields. Therefore, our scheme is post-quantum in nature since there is no efficient quantum algorithm to solve ill-posed inverse problems in polynomial time.

Remark 2.1 (Requirement of a synchronized error generator)**.** For a symmetric encryption scheme, both sender and receiver should have the same secret key. In our case, the secret key is error. Consequently, both the parties should have a synchronized error generator that generates error uniformly at random (or following certain distribution).

2.6. Advantages and disadvantages of our scheme along with comparison analysis

In this subsection, we briefly discuss the advantages and disadvantages of our scheme. We also perform a comparison analysis and compare our scheme with various other well-known schemes.

2.6.1. Advantages

- **Infinite dimensional structure**: Working in $L^2[0,1]$ offers no direct quantum speedups and the scheme is resistant to attacks based on Gröbner basis and algebraic attacks (see [6]). Further, our scheme offers operator-theoretic hardness since the corresponding inverse problem is highly ill-posed.
- **Implementation**: The encryption scheme can be interpreted as

  $$\text{hidden linear system} + \text{noise}.$$

  This is very easy to implement.
- **Flexibility**: One can obtain many variants of the scheme by choosing different encodings, compact operator and noise shapes.
- **Large error space**: The error space can be easily generated with the approach discussed in subsection 2.1. Additionally, its size can be easily changed by changing N. This renders brute force search infeasible and also offers analytic flexibility in selecting noise.
- **Potential post-quantum security**: As there are no known efficient quantum algorithms for inverting a compact operator with noisy data, our scheme offers post-quantum security.

2.6.2. Disadvantages

- **No formal hardness reduction**: Although our scheme is CCA2 secure, but unlike well-known schemes such as LWE [20], classic McEliece [4] etc., there is no proof of its hardness in the worst case. This may be taken up as a future task.
- **Performance concerns**: It may be costly to simulate the discretize version of infinite dimensional operators. Accordingly, key and ciphertext sizes depend on discretization.

2.6.3. Comparison analysis

In this subsection, we compare our scheme with several other well-known encryption schemes available in literature. Specifically, we consider RSA and ECC (elliptic curve cryptography) schemes [13, 17], scheme based on learning with error problem [20], McEliece scheme [4], group ring based schemes [14, 15, 18], Advance encryption standard (AES) [21], symmetric encryption scheme based on Reed-Solomon codes (SERSC) [11], ChaCha20 [1] and Salsa20 [2]. The comparison is given in Table 2. The abbreviations used in Table 2 are as follows: DLP means discrete logarithm problem, ECDLP means elliptic curve DLP, GRDLP means group ring DLP, SPN means substitution-permutation network, ARX means add-rotate-XOR, PQ denotes post-quantum, CS denotes computationally secure (it means that is not possible to break the scheme in polynomial time with the best available resources).

It is clear from Table 2 that the main highlight of our scheme is that it is based on a different Mathematical setup which is not yet explored. Overall, the associated Mathematical problem with our scheme is believed to be hard since it is analogous to LWE problem.

Table 2. Comparison of our scheme with several other schemes.

Scheme	**Structure**	**Security Basis**	**PQ?**	**Our Scheme vs. this scheme**
RSA	Number theory	Factoring	×	Our scheme is PQ-secure
ECC	Number theory	ECDLP	×	Our scheme is PQ-secure
LWE	Matrix algebra	Lattice hardness	✓	Security of LWE scheme is very well studied whereas the security of our scheme relies on a hard problem problem analogous to LWE but it needs more scrutiny
McEliece	Coding theory	Decoding	✓	Security depends on NP-hard problem which is not the case with our scheme
Group ring	Group algebra	GRDLP	✓	This require more sophisticated implementation in comparison to our scheme
AES-128	SPN	CS	×	AES is extremely fast in comparison to our scheme but our scheme offers different mathematics
SERSC	Coding theory	Decoding	✓	Based on a technique to construct a binary code from a non-binary Reed-Solomon code. Mathematics of both the schemes is different
Salsa20	ARX	CS	×	This is very fast in software and incorporates basic mathematical operations unlike our scheme
ChaCha20	ARX	CS	×	This scheme is a variant of Salsa20
Our Scheme	Functional analysis	Ill-posedness	✓	–

3. Public key encryption scheme based on ill-posed problems

In this section, we build on the work of previous section and propose a public key encryption (PKE) scheme based on ill-posed problems. Specifically, we integrate the symmetric encryption scheme proposed in the previous section with CRYSTALS-Kyber key encapsulation mechanism (KEM) [5].

3.1. Key generation

We run CRYSTALS-Kyber KEM *key generation algorithm* to generate a public/private key pair. We denote it by pk and sk, respectively.

3.2. Encryption

Let $\mu \in \{0,1\}^t$ be the message to be encrypted. We run CRYSTALS-Kyber KEM *encaps algorithm* to generate a key $\mathcal{K}'$ (of length 256 bits) and the corresponding ciphertext $\mathcal{C}_1$. Further, we instantiate XOF function [19] on $\mathcal{K}'$ to get a key $\mathcal{K}$ of desired length to be used for symmetric encryption scheme S described in section 2. Finally, we encrypt μ using the encryption function of the scheme S to get ciphertext $\mathcal{C}_2$. The final ciphertext is $(\mathcal{C}_1, \mathcal{C}_2)$.

3.3. Decryption

After receiving $(\mathcal{C}_1, \mathcal{C}_2)$, the decryptor first run CRYSTALS-Kyber KEM *decaps algorithm* with input $\mathcal{C}_1$ to get the key $\mathcal{K}'$. Then, the decryptor runs XOF function on $\mathcal{K}'$ to get the key $\mathcal{K}$. Finally, using the decryption function of the scheme S on ciphertext $\mathcal{C}_2$ with key $\mathcal{K}$, we get the message μ.

3.4. Security analysis

The security of our PKE clearly depends on the security of Kyber KEM and symmetric encryption scheme S. The security of Kyber KEM depends on lattice problem, i.e., module-LWE (module learning with errors), whose hardness is very well understood (see [5]). The security of S is already discussed, i.e., it is CCA2 secure. Consequently, it follows from [5] that our PKE is IND-CCA2 secure (indistinguishability under adaptive chosen ciphertext attack), which is the golden security standard.

4. Discussion

We have unveiled a canonical analogy between post-quantum lattice based learning with error problems and ill-posed inverse problems. Precisely, we have shown that LWE problem is a special case of solving ill-posed inverse problems (subsection 1.4). Motivated from this fact, we have proposed two encryption schemes. The first one is symmetric and other one is asymmetric (or PKE) based on ill-posed inverse problems. We also thoroughly discussed the security of these schemes. In future, this work can be extended in a number of ways. The first one is to look at the impact of regularization techniques on the security of these schemes. The second one is to look at the efficient ways of sampling errors in (2.3).

Acknowledgement

We sincerely thank the reviewer and editor for their careful reading of the manuscript and for the constructive comments, which have helped us significantly improve the quality and clarity of the paper.

References

[1] D. J. Bernstein: *ChaCha, a Variant of Salsa20*, in: Workshop Record of SASC, vol. 8, 1, 2008, pp. 3–5.

[2] D. J. Bernstein: *The Salsa20 Family of Stream Ciphers*, in: New Stream Cipher Designs: The eSTREAM Finalists, Berlin: Springer, 2008, pp. 84–97.

[3] D. J. Bernstein, J. Buchmann, E. Dahmen: *Post-Quantum Cryptography*, Berlin: Springer, 2009, doi: 10.1007/978-3-540-88702-7.

[4] D. J. Bernstein, T. Chou, T. Lange, et al.: *Classic McEliece: Conservative Code-Based Cryptography*, NIST Post-Quantum Cryptography Submission, 2017.

[5] J. Bos, L. Ducas, E. Kiltz, et al.: *CRYSTALS-Kyber: A CCA-Secure Module-Lattice-Based KEM*, in: 2018 IEEE European Symposium on Security and Privacy, IEEE, 2018, pp. 353–367, doi: 10.1109/EuroSP.2018.00032.

[6] C. Cid, R.-P. Weinmann: *Block Ciphers: Algebraic Cryptanalysis and Gröbner Bases*, in: Gröbner Bases, Coding, and Cryptography, Berlin, Heidelberg: Springer, 2009, pp. 307–327, doi: 10.1007/978-3-540-93806-4_17.

[7] L. Ducas, E. Kiltz, T. Lepoint, et al.: *CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme*, IACR Transactions on Cryptographic Hardware and Embedded Systems (2018), pp. 238–268, doi: 10.13154/tches.v2018.i1.238-268.

[8] H. W. Engl, M. Hanke, A. Neubauer: *Regularization of Inverse Problems*, Kluwer Academic Publishers, 2000.

[9] S. Fluhrer: *Reassessing Grover's Algorithm*, Cryptology ePrint Archive, Report 2017/811, 2017.

[10] G. Fuchsbauer: *An Introduction to Probabilistic Encryption*, Osječki Matematički List 6.1 (2006), pp. 37–44.

[11] C. Hannusch: *A Symmetric Cryptosystem Based on Reed–Solomon Codes*, International Journal of Latest Engineering Research and Applications 4.11 (2019), pp. 9–12.

[12] H. Heys: *A Tutorial on Linear and Differential Cryptanalysis*, Cryptologia 26.3 (2002), pp. 189–221, doi: 10.1080/0161-110291890885.

[13] N. Koblitz, A. Menezes, S. Vanstone: *The State of Elliptic Curve Cryptography*, Designs, Codes and Cryptography 19.2 (2000), pp. 173–193, doi: 10.1023/A:1008354106356.

[14] S. Kumar, G. Mittal, S. Kumar: *A Secure Key Authentication Scheme for Cryptosystems Based on DLP in Group Ring*, Annales Mathematicae et Informaticae 60 (2024), pp. 75–92, doi: 10.33039/ami.2024.03.004.

[15] S. Kumar, G. Mittal, A. Yadav: *A Novel and Provably Secure Identity-Based Blind Signature Scheme for Online Transactions*, Sādhanā 50.2 (2025), pp. 1–12, doi: 10.1007/s12046-025-02694-1.

[16] R. LaPierre: *Shor Algorithm*, in: Introduction to Quantum Computing, Springer Nature Switzerland, 2025, pp. 189–205, doi: 10.1007/978-3-031-90731-9.

[17] A. Menezes, P. Van Oorschot, S. Vanstone: *Handbook of Applied Cryptography*, CRC Press, 2018.

[18] G. Mittal, S. Kumar, S. Kumar, S. Mittal: *A Novel and Efficient Undeniable Signature Scheme Based on Group Ring*, Soft Computing 28.23 (2024), pp. 13053–13070, doi: 10.1007/s00500-024-10325-w.

[19] National Institute of Standards and Technology: *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*, Federal Information Processing Standards Publication 202, 2015, doi: 10.6028/NIST.FIPS.202.

[20] O. Regev: *On Lattices, Learning with Errors, Random Linear Codes, and Cryptography*, Journal of the ACM 56.6 (2009), pp. 1–40, doi: 10.1145/1568318.1568324.

[21] D. Salent: *Advanced Encryption Standard*, Rivier Academic Journal 6.2 (2010), pp. 1–140.