SUST Journal of Science and Technology

journal homepage: https://journals.sust.edu

SUST JOURNAL OF SCIENCE AND TECHNOLOGY

Quantum-Resistant Diffie-Hellman Key Exchange Protocol

A. K. M. Fakhrul Hossain*

Department of Computer Science and Engineering, Shahjalal University of Science and Technology, Sylhet – 3114, Bangladesh

ARTICLE INFO

Keywords:

Diffie-Hellman Key Exchange Protocol
Authentication
Man-in-the-Middle Attack
Learning With Errors
Quantum Threats
Lattice Based Cryptography
Post-Quantum Cryptography
Dolev-Yao model
CRYSTALS-Dilithium

ABSTRACT

The necessity of a secure key exchange protocol arises from the critical need to establish encrypted communication over an untrusted network. Over time, a multitude of key exchange mechanisms have been developed to counteract adversarial threats. The Diffie-Hellman key exchange protocol (DHKE) is one of the most widely used protocols for symmetric key sharing. However, this protocol exhibits certain inherent limitations that attackers may exploit. It lacks authentication mechanism and is susceptible to Man-in-the-Middle (MITM) attacks and quantum attacks. To mitigate these vulnerabilities, we have designed a hybrid key exchange protocol combining DHKE with Learning With Errors (LWE), a lattice-based post-quantum primitive. This proposed protocol provides authentication via a Public Key Infrastructure (PKI) together with CRYSTALS-Dilithium digital signature, resilience against MITM attacks, and robustness against classical and quantum threats. We have done a security analysis using the Dolev-Yao threat model, extended to quantum-equipped attackers, and showed that the protocol achieves mutual authentication, session key secrecy, and forward secrecy under the hardness of LWE and DHKE. Lastly, we provided detailed parameter recommendations based on NIST standards and shed light on side-channel attacks.

1. Introduction

To communicate with someone over the internet, we must use the public network. Any public channel is an open target for the adversaries. Hence, the encrypted channel comes into play. But for that, the communicating parties need to have either a public-private key pair or a symmetric key. Due to the latency and computational overhead, symmetric keys are appropriate for back-and-forth communication. However, exchanging a symmetric key between the users is challenging and needs a secure key exchange protocol.

The Diffie-Hellman (DHKE) [1] is a widely used key exchange protocol where any two unknown parties can establish a shared symmetric key for further encrypted communication. Even though it was proposed in 1976, its classical version and some variants are still widely used in numerous protocols and modules in the web.

The DHKE is an elegant protocol in the sense that it involves forward secrecy between the parties, that is, both of the parties forward its private key in such a secret manner that no one can figure it out, not even the person contacting to, but eventually they'll conform to a similar secret key by merging these obscure parts.

The strength of this protocol comes from the difficulty of solving discrete logarithm problems (DLP) using classical approaches. Nevertheless, a variant of Shor's algorithm can solve the DLP in a feasible time [2]. Hence, this poses a severe threat to this legacy protocol and the systems utilizing DHKE are on the first target list of the imminent quantum attack in the future. Moreover, due to the lack of authenticity, this protocol is prone to (Man-in-the-Middle) MITM attacks [3].

This scenario motivates us to design an authentic quantum-resistant key exchange protocol. Our key contributions are: (1) a formal Dolev-Yao threat model extended to a quantum-capable adversary; (2) a hybrid

* Corresponding author.

E-mail address: fakhrul-cse@sust.edu (A. K. M. F. Hossain)

https://doi.org/10.63512/sustjst.2025.1006

Received 16 December 2025; Received in revised form 12 May 2026; Accepted 10 August 2026

DHKE-LWE protocol hides DLP parameters under LWE hardness against the Shor's algorithm; (3) an authentication mechanism using PKI-certified LWE public keys and CRYSTALS-Dilithium post-quantum digital signature; (4) standard pseudocode based protocol representation; (5) concrete parameter selection guidance according to NIST post-quantum standards; and (6) discussion of side-channel attack mitigation.

Our rationale behind combining DHKE with LWE is as follows: (1) the DHKE provides forward secrecy for each session using the ephemeral exponents, which is not achievable using the digital certificate anchored long-term public keys in LWE; (2) the hybrid protocol offers a feasible upgrade path to post-quantum protocol for systems using classical DHKE (TLS 1.2/1.3, IKEv2, SSH, and many industrial protocols); (3) DHKE exposes DLP parameters over the transmission channel, making them vulnerable to Shor's algorithm. LWE hides the discrete logarithm primitives; (4) LWE keys, binding with a digital certificate, ensure authentication; (5) the combined hardness guarantees stronger security than either alone, and the protocol remains safe as long as at least one of DH or LWE is unbroken which aligns with NIST SP 800-227 guidance on hybrid key establishment [27].

2. Background Studies

2.1. Diffie-Hellman Key Exchange Protocol: Strengths and Weaknesses

At the outset, the two communicating entities, let Alice and Bob, publicly agree to use a large prime number P and a generator g in the prime-group for further processing. After that, Alice picks a random number a and Bob picks a random number b such that $(a, b) \in \mathbb{Z}_P$ where $\mathbb{Z}_P$ is the residue set of P and ranges from 0 to $P-1$.

Now, Alice and Bob calculate $A = g^a \bmod P$, and $B = g^b \bmod P$ respectively. Next, Alice and Bob reciprocally share A and B among themselves. Finally, Alice calculates the shared key as $K_a = B^a \bmod P$ and likewise Bob calculates it as $K_b = A^b \bmod P$. Here, $K_a = B^a \bmod P = (g^b)^a \bmod P = (g^a)^b \bmod P = A^b \bmod P = K_b$. Therefore, the final symmetric key is $K_a = K_b = g^{ab} \bmod P$.

The main strength of this protocol comes from the hardness of solving the DLP, that is, for sufficiently large prime numbers (of 2048 or above bits), no known classical algorithm can find $Z = X^Y \bmod P$, in a feasible time, even if Z, X and P are known. Moreover, forward secrecy of the symmetric key makes it an elegant approach to be used as a key distribution protocol without the necessity of any prior communication.

These advantages are incorporated by the parties to widely use the protocol. However, due to the lack of any authentication mechanism, this protocol is susceptible to MITM attacks. Such an attack is shown here [3] where the adversary can easily spoof the parties and impersonate them. The victim users can't even identify whether they are communicating with the attacker or not. Such a scenario is shown below:

Let Alice and Bob, with no prior knowledge of each other, want to establish a shared key between them using DHKE. Both of them decided to use the prime P and generator g at the setup phase. Additionally, Alice and Bob picked two random numbers x and y respectively where $(x, y) \in \mathbb{Z}_P$. As they're trying to communicate in a public channel, adversaries (let Mallory) are present there and can impose both passive and active attacks.

Now, to impose an MITM attack, Mallory also picks two random numbers s and t where $(s, t) \in \mathbb{Z}_P$. s will be used to attack Bob and t will be used for Alice.

As a regular exchange, Bob calculates $X = g^x \bmod P$ and on the other side Alice also calculates $Y = g^y \bmod P$. They send X and Y to the public network to share it with each other. This time, Mallory captures both of their data packets and sends $S = g^s \bmod P$ to Bob and $T = g^t \bmod P$ to Alice. As there is no mechanism to verify the authenticity of the source of the data, neither Bob nor Alice can assure the identity of the sender. Bob assumes S comes from Alice and Alice assumes the same for Bob regarding T.

Now, Bob and the attacker calculate their shared secret key as $K_{xs} = g^{xs} \bmod P$. For Alice and the attacker, it is calculated as $K_{yt} = g^{yt} \bmod P$ as shown in Figure 1. Bob and Alice, both of them thinks that they are communicating with each other, but the fact is that, Mallory spoofed both of their identities and is pretending to be the counterpart with each of them. He can easily decrypt and read what Alice or Bob sends. Also, he can send new message to each other fully breaking confidentiality, integrity and authenticity.

Even after being an excellent tool for key exchange with the power of forward secrecy and high computational attack resistance due to DLP, DHKE fails to play a secure role against an active MITM attack due to its inherent inability to ensure authenticity.

Additionally, though DLP is classically a hard problem of exponential time complexity, which makes it infeasible for any computer to break in practical time, Shor's algorithm reduces the complexity in polynomial time using his groundbreaking algorithm [4]. This algorithm can find the private key (the exponent), given the public key (the modulus and the generator). This threat makes the DHKE protocol quantum vulnerable.

2.2. Lattice Based Cryptography: A Post Quantum Approach

Lattices are infinitely repeated points in $\mathbb{R}^n$ space of the form $x_1 v_1 + x_2 v_2 + \cdots + x_m v_m$, where $(x_i)_{i \le m} \in \mathbb{Z}^n$ and $\{v_1, v_2, \cdots, v_m\}$ is linearly independent in $\mathbb{R}^n$. Lattices involve complex and hard problems that are difficult to solve, even using existing quantum algorithms. Recently, even the NIST is moving towards lattice-based cryptography for the post-quantum era to safeguard against potential quantum attacks [5].

Some hard problems like, shortest vector problem (SVP) (finding the shortest non-zero vector in the lattice space, which is quite hard when the dimension is increased), the closest vector problem (CVP) (finding the lattice point which is closest to some given point), form the basis of lattice-based cryptography and make it a better candidate to shield against quantum threats.

Regarding the strength of SVP and CVP, both of them are computationally NP-hard problems [6, 7]. Even the approximated SVP or CVP within small factors remains computa-

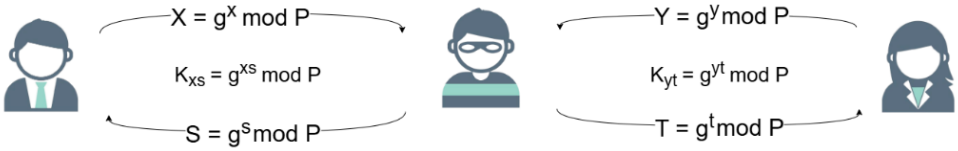

Figure 1. Man-in-the-Middle Attack on Diffie-Hellman Key Exchange Protocol

tionally beyond polynomial bounds unless NP = RP. This hardness makes it a strong candidate for modern cryptography.

2.3. Learning With Errors (LWE): Strengths and Possibilities

LWE is a form of computationally challenging lattice-based problem. It provides a foundation for numerous recently proposed cryptosystems. The challenge involves finding a secret vector from a set of linear equations that has been intentionally corrupted by adding random errors.

For any random matrix $\mathbf{A}$ in $\mathbb{Z}^{m\times n}/(x^n+1)$ and a chosen secret vector $\mathbf{s}$ in the finite field $\mathbb{Z}_q^n/(x^n+1)$, the LWE problem involves pairs $(\mathbf{A}, \mathbf{b})$, where $\mathbf{b} = \mathbf{A}\cdot\mathbf{s} + \mathbf{e} \bmod q$ and $\mathbf{e}$ is a random error vector from a distribution that is small enough and strictly less than $q/4$. The ultimate target is to find the secret $\mathbf{s}$, provided many public pairs from $(\mathbf{A}, \mathbf{b})$ [8].

Let Alice wants to send a message $M = m_0 m_1 \cdots m_n$ to Bob. Bob has chosen a random matrix $\mathbf{A}$ such that $\mathbf{A} \in \mathbb{Z}_q^{m\times n}/(x^n+1)$, that is $\mathbf{A}$ is an $n\times k$ matrix having each entry with a random polynomial modulo x^n+1 and coefficients being module q. He also chose a secret vector $\mathbf{s} \in \mathbb{Z}_q^n/(x^n+1)$. Additionally, he picks a random error vector $\mathbf{e}$ sampled from a discrete Gaussian or Binomial distribution and $\mathbf{e} \in \mathbb{Z}_q^m/(x^n+1)$ and $|\mathbf{e}_i| \le \lfloor q/4 \rfloor$. Then, he calculates the vector $\mathbf{b} = (\mathbf{A}\cdot\mathbf{s} + \mathbf{e}) \bmod q$. Bob sets $(\mathbf{A}, \mathbf{b})$ as the public key and keeps $\mathbf{s}$ as the secret key.

Now, for sending the message M, Alice converts the message into a polynomial modulo x^n+1, that is $\mu(x) = (m_0 + m_1 x + m_2 x^2 + \cdots + m_n x^{n-1})/(x^n+1)$. Now, for picking random rows of polynomials, Alice picks a random vector $\mathbf{d} \in \{0,1\}^m$. Finally, she calculates the ciphertext C as $C = \mathbf{d}\cdot\mathbf{b} + \lfloor q/2 \rfloor \mu(x)$ and sends $(C, \mathbf{d})$ to Bob.

To decrypt the ciphertext, using the private key and cipher, Bob calculates

$$\begin{aligned} M' &= C - \mathbf{d}\cdot\mathbf{A}\cdot\mathbf{s} \\ &= \mathbf{d}\cdot\mathbf{b} + \lfloor q/2 \rfloor \mu(x) - \mathbf{d}\cdot\mathbf{A}\cdot\mathbf{s} \\ &= \mathbf{d}\cdot\mathbf{A}\cdot\mathbf{s} + \mathbf{d}\cdot\mathbf{e} + \lfloor q/2 \rfloor \mu(x) - \mathbf{d}\cdot\mathbf{A}\cdot\mathbf{s} \\ &= \lfloor q/2 \rfloor \mu(x) + \mathbf{d}\cdot\mathbf{e}. \end{aligned}$$

Now, the m_i-th character is 0 if $M_i' \approx 0$ and $m_i = 1$ if $M_i' \approx q/2$.

LWE offers strong theoretical hardness. Even with polynomial modulus, solving the LWE problem is at least as hard as standard worst-case lattice problems, both classically [7, 9] and also using quantum algorithms [8]. The best known classical algorithm, involving lattice reduction by the BKZ algorithm for solving the LWE problem has sub-exponential time complexity of approximately $2^{0.292d}$, where d is the block size parameter [10]. Moreover, using quantum-classical hybrid algorithms may be promising, but they are mostly heuristic and require significant quantum resources [11]. This makes LWE robust against any classical or quantum threat and hence it emerges as a dependable means for future cryptography.

3. Related Works

As discussed earlier, DHKE protocol is an excellent and promising means for key exchange. However, some of its shortcomings like, the absence of any authentication mechanism, quantum vulnerability, and being prone to MITM. There have been numerous significant works to improve the DHKE protocol to mitigate its weaknesses.

Tefra et al. proposed a way to mitigate eavesdropping on the DHKE protocol in their work [12]. According to their proposal, half of the key will be shared in the public channel, the same as the traditional DHKE. The other half will be shared in a secure and alternative trusted channel like, email or over the phone. They claim that this will ensure the integrity and authenticity of the key. Main drawback of this approach is that they first need an additional secure medium. If there is a secure medium, then there is no need for the first step of sharing the partial key in an insecure public medium.

Rakel et al. designed an alternative approach for DHKE using elliptic curves instead of large integers in their work [13]. This provides similar security but with a much shorter key and hence ensures stronger protection and resource efficiency. Nevertheless, the elliptic curve DHKE protocol still incorporates the DLP and can be easily attacked using modified Shor's algorithm for solving the elliptic curve discrete logarithm problem [2].

Mahmood Khalel proposed a DHKE protocol adding authenticity using zero-knowledge proof (ZKP) [3]. His designed protocol is divided into two phases. The first phase is similar to the basic DHKE and is susceptible to MITM attacks. The second phase uses ZKP but still uses the DLP as the base, and so can be easily broken by Shor's algorithm [2]. Moreover, the way ZKP is used in this proposal, cannot stop the active MITM attack, and both of the communicating parties' identities can be spoofed.

Georgios proposed a quantum DHKE protocol in his work [14], where he designed a quantum one-way function for bijective mapping of large numbers into some homogenous coherent states. His work is promising, but measuring quantum states might introduce additional errors because of an inherent property of the quantum system. Moreover, the communicating parties need to have a quantum setup to use this protocol, which is, however, costly.

Dima et al. introduced a novel approach to a generalized version of DHKE on neutrosophic and split-complex

integers [15]. Their approach is similar to the classical DHKE, but they have used the neutrosophic number and complex number in their approach. This somewhat increases the search domain for classical attacks, but nothing much changes for quantum threats. Their approach yet is a DLP pattern and is vulnerable to Shor's algorithm [2].

4. Authentication Mechanism

A critical requirement for MITM resistance is that public keys are need to be bound to verifiable identities and a digital signature mechanism is incorporated. Our protocol assumes the existence of a Public Key Infrastructure (PKI) in which each party's LWE public key is certified by a trusted Certificate Authority (CA). Each party's LWE public key tuple $(\mathbf{A}, \mathbf{b}, P, g)$ is associated with a CA-issued certificate binding the key to the entity's verified identity.

Furthermore, each transmitted ciphertext key is signed using CRYSTALS-Dilithium [26], a post-quantum digital signature scheme, standardized by the NIST. Upon receiving any ciphertext, each party verifies the signature using the sender's certified public key before accepting. This two-layer authentication — PKI certificate for key binding and digital signature for message authentication — ensures that both MITM attacks and impersonation attacks are mitigated even by the quantum-equipped adversaries.

5. Threat Model: Dolev-Yao with Quantum Extension

We adopt the Dolev-Yao threat model [25], the standard symbolic attacker model for formal analysis of cryptographic protocols, to design threat modelling for our one. Under this model, the adversary is present within the network, and he can intercept every message transmitted by a valid participant. He may then decide whether to forward it, modify it, drop it, or inject a new message replacing it. All public keys and protocol parameters are available to the attacker. Cryptographic primitives are treated as ideal black boxes under the assumption that, ciphertexts are indistinguishable without the corresponding key, and digital signatures are unforgettable and verifiable. This abstraction helps to differentiate protocol-level vulnerabilities from implementation-level flaws, making it the suitable model for our analysis.

5.1. Preliminaries and Notations

We fix the following notation to be used throughout the paper.

$\mathcal{A} = \{\text{Alice}, \text{Bob}, I\}$ – the set of principals, where I denotes the adversary/intruder.

$\mathcal{N} = \{n_1, n_2, ...\}$ – the set of nonces.

$\mathcal{K} = \{k_1, k_2, ...\}$ – the set of cryptographic keys.

For each principal, $A \in \mathcal{A}$, we denote:

$sk_{CD}(A)$: CRYSTALS-Dilithium secret key for signing,

$vk_{CD}(A)$: CRYSTALS-Dilithium signature verification key, public,

$sk_{lwe}(A)$: LWE secret key,

$pk_{lwe}(A) = (\mathbf{A}_a, \mathbf{b}_a, P, g)$: Public key.

For symmetric keys, we assume: $inv(k) = k$

For asymmetric pairs: $inv(vk(A)) = sk(A)$, $inv(sk(A)) = vk(A)$.

The message term grammar over signature $F = \{\text{pair}, \text{sign}, \text{ver}, \text{hash}, \text{lwe\_enc}, \text{lwe\_dec}\}$ is defined as follows:

$t ::= a \mid n \mid k$	(principals, nonces, keys)
$\mid \langle t_0, t_1 \rangle$	(pairing/concatenation)
$\mid \text{lwe\_enc}(t, pk_{lwe}(A))$	(LWE encryption of t using A's public key)
$\mid \text{lwe\_dec}(t, sk_{lwe}(A))$	(LWE decryption of t using A's secret key)
$\mid \text{sign}(t, sk_{CD}(A))$	(Dilithium sign on t by A)
$\mid \text{ver}(t, t', vk_{CD}(A))$	(signature verification: true iff $t = \text{sign}(t', sk_{CD}(A))$)
$\mid \text{hash}(t)$	(standard hash of t)

The standard equation theory E over F captures the constructor-destructor relationships as below:

$$\text{lwe\_dec}(\text{lwe\_enc}(t, pk_{lwe}(A)), sk_{lwe}(A)) = t$$

$$\text{ver}(\text{sign}(t, sk_{CD}(A)), t, vk_{CD}(A)) = \text{true}$$

The derivation relation $X \vdash t$ (read: the adversary can derive t from the known terms X). The adversary can apply any constructor or destructor or terms in their knowledge set. The derivability problem – deciding whether $X \vdash t$ – is decidable in polynomial time for the standard Dolev-Yao term algebra [25].

Some derivation rules can be found easily from the grammar. Like, $X \vdash t$, $X \vdash pk_{lwe} \Rightarrow X \vdash \text{lwe\_enc}(t, pk_{lwe})$ means LWE encryption of the term t can be derived with the LWE public key, $X \vdash t$, $X \vdash sk_{CD} \Rightarrow X \vdash \text{sign}(t, sk_{CD})$ means signing can be done on t with known secret key using Dilithium. Other inference rules can intuitively found from the given grammar over F. And, all the public parameters are by default derivable.

5.2. Adversary Capabilities

The DY adversary I is assumed to be the most powerful possible active network attacker. Formally, I maintains a monotonically growing knowledge set $\Omega \subseteq T$, initialised to all public information:

$$\Omega_0 = \{a \mid a \in \mathcal{A}\} \cup \{vk_{CD}(A) \mid A \in \mathcal{A}\} \cup \{pk_{lwe}(A) \mid A \in \mathcal{A}\} \cup \{P, g\}$$

The adversary can perform the following operations at any time during a protocol execution:

(C1) Intercept: For any message m sent by a valid principal A to B, I receives m before B and adds it to Ω: $\Omega \leftarrow \Omega \cup \{m\}$. The message may or may not be forwarded to B.

(C2) Inject: For any term t such that $\Omega \vdash t$, I may send t to any principal on any channel, appearing to come from any principal $A \in \mathcal{A}$.

(C3) Replay: For any term $t \in \Omega$ (previously intercepted), I may resend t to any principal at any time.

(C4) Delay: I may delay delivery of any message for any finite number of steps.

(C5) Drop: I may suppress any message entirely, preventing it from reaching its intended recipient.

(C6) Compose: Given $\Omega \vdash t_0$ and $\Omega \vdash t_1$, I can derive $\langle t_0, t_1 \rangle$, $\text{lwe\_enc}(t_0, pk_{lwe}(A))$, $\text{sign}(t_0, sk_{CD}(I))$, $\text{hash}(t_0)$, and any other term derivable from Ω under the equational theory E.

(C7) Quantum computation: We extend the classical DY model to a quantum-capable adversary: I is additionally assumed to have access to a fault-tolerant quantum computer, capable of running Shor's algorithm [4] to solve the Discrete Logarithm Problem (DLP) in polynomial time. This extension is denoted DY_{QC}.

I cannot break the LWE encryption $\mathrm{lwe\_enc}(t_0, pk_{lwe}(A))$ without $sk_{lwe}(A)$, cannot forge a Dilithium signature $\mathrm{sign}(t, sk_{CD}(A))$ without $sk_{CD}(A)$, and cannot invert $\mathrm{hash}(t)$. These are the perfect cryptography assumptions for LWE encryption and Dilithium signatures, justified by the computational hardness of LWE [8, 9] and Module-LWE [26], respectively. Shor's algorithm is inapplicable to LWE because the noise in the LWE construction breaks the periodicity required by Shor's period-finding subroutine.

5.3. Protocol Under DY_{QC}

We now describe the honest execution of our protocol in the DY narration. Let n_a and n_b be fresh session nonces generated by Alice and Bob respectively to prevent replay attacks. Let $cert_a$ and $cert_b$ be certificates issued by CA, binding each party's LWE public key to their identity.

Step 0 (Setup): Alice and Bob each obtain CA-issued certificates for their LWE public keys. Both verify each other's certificates against the trusted CA public key $vk_{CD}(CA)$.

$$\text{Alice} \to \text{Bob} : cert_a = \langle pk_{lwe}(\text{Alice}), vk_{CD}(\text{Alice}), \text{Alice}\rangle_{sk(CA)}$$

$$\text{Bob} \to \text{Alice} : cert_b = \langle pk_{lwe}(\text{Bob}), vk_{CD}(\text{Bob}), \text{Bob}\rangle_{sk(CA)}$$

Step 1 (Alice's Transmission): Alice generates fresh nonce n_a, computes $X = g^x \bmod P$, encrypts X under Bob's LWE public key, signs the ciphertext with her Dilithium key and sends to Bob:

$$new(n_a); \quad X := g^x \bmod P;$$

$$C_X := \mathrm{lwe\_enc}(\langle X, n_a, \text{Alice}\rangle, pk_{lwe}(\text{Bob}))$$

$$\sigma_a := \mathrm{sign}(\langle C_X, n_a, \text{Bob}\rangle, sk_{CD}(\text{Alice}))$$

$$\text{Alice} \to \text{Bob} : \langle C_X, n_a, \sigma_a, cert_a\rangle$$

Step 2 (Bob transmits): Bob generates new nonce n_b, computes $Y = g^y \bmod P$, and sends to Alice likewise:

$$new(n_b); \quad Y := g^y \bmod P;$$

$$C_Y := \mathrm{lwe\_enc}(\langle Y, n_b, \text{Bob}\rangle, pk_{lwe}(\text{Alice}))$$

$$\sigma_b := \mathrm{sign}(\langle C_Y, n_b, \text{Alice}\rangle, sk_{CD}(\text{Bob}))$$

$$\text{Bob} \to \text{Alice} : \langle C_Y, n_b, \sigma_b, cert_b\rangle$$

Step 3 (Key construction): Upon receipt, each party first verifies the certificate and Dilithium signature, then decrypts and computes the shared key:

Bob receives $\langle C_X, n_a, \sigma_a, cert_a\rangle$:

- $\mathrm{ver}(cert_a, \langle pk_{lwe}(\text{Alice}), vk_{CD}(\text{Alice}), \text{Alice}\rangle, vk_{CD}(CA)) = \text{true}$
- $\mathrm{ver}(\sigma_a, \langle C_X, n_a, \text{Bob}\rangle, vk_{CD}(\text{Alice})) = \text{true}$
- $X := \mathrm{lwe\_dec}(C_X, sk_{lwe}(\text{Bob}))$
- $K_b := X^y \bmod P$

Alice receives $\langle C_Y, n_b, \sigma_b, cert_b\rangle$:

- $\mathrm{ver}(cert_b, \langle pk_{lwe}(\text{Bob}), vk_{CD}(\text{Bob}), \text{Bob}\rangle, vk_{CD}(CA)) = \text{true}$
- $\mathrm{ver}(\sigma_b, \langle C_Y, n_b, \text{Alice}\rangle, vk_{CD}(\text{Bob})) = \text{true}$
- $Y := \mathrm{lwe\_dec}(C_Y, sk_{lwe}(\text{Alice}))$
- $K_a := Y^x \bmod P$

Result: $K_a = K_b = g^{xy} \bmod P$

5.4. Security Goals Achieved Under DY_{QC}

We provide argument that the following security goals are achieved by the protocol against the DY_{QC} adversary I:

G1 – Secrecy of the Shared Key: After the successful completion of all the phases between honest principals, the shared key $K = g^{xy} \bmod P$ is not derivable by I. That is, for all protocol executions τ, consistent with the DY_{QC} model, $\Omega_\tau \vdash K$ must not hold where Ω_τ is the knowledge of the attacker after τ executions.

The adversary I observes only $\langle C_X, n_a, \sigma_a, cert_a\rangle$ and $\langle C_Y, n_b, \sigma_b, cert_b\rangle$ on the channel. $C_X = \mathrm{lwe\_enc}(\langle X, n_a, \text{Alice}\rangle, pk_{lwe}(\text{Bob}))$ hides X under LWE encryption. By the perfect cryptography assumption for LWE (justified by the hardness of LWE [8, 9]), $\Omega \vdash X$ requires $sk_{lwe}(\text{Bob})$, which I does not possess. Without $X = g^x \bmod P$, I cannot compute $K = g^{xy} \bmod P$ even with the quantum DLP oracle (C7), as the DLP parameter X is not accessible. Hence G1 holds.

G2 – Authentication: Any message received by Bob expected to be sent by Alice must satisfy $\mathrm{ver}(\sigma_a, \langle C_X, n_a, \text{Bob}\rangle, vk_{CD}(\text{Alice})) = \text{true}$. By the standardized cryptography assumption for Dilithium, I cannot produce a valid σ_a without $sk_{CD}(\text{Alice})$. Hence I cannot successfully impersonate Alice to Bob, and vice-verca, and hence G2 holds. The certificate verification step further ensures that $vk_{CD}(\text{Alice})$ is genuinely bound to Alice's identity.

G3 — Replay resistance: Each $\sigma_a = \mathrm{sign}(\langle C_X, n_a, \text{Bob}\rangle, sk_{CD}(\text{Alice}))$ binds the session to a fresh nonce $n_a \in \mathcal{N}$ generated by $new(n_a)$. A replayed σ_a from any previous session would not match ($n'_a \neq n_a$) for the current session. The receiving principal verifies the nonce with the current session context; upon mismatch, the session is aborted. Hence G3 holds.

G4 — Forward secrecy: Exponents x and y are ephemeral and random: generated freshly per session and never transmitted or stored beyond the session. Even if I later obtains $sk_{CD}(\text{Alice})$, $sk_{lwe}(\text{Alice})$, or $sk_{lwe}(\text{Bob})$, the values x and y are no longer recoverable (they are not derivable from Ω by any inference rule over F). Hence, G4 holds according to DHKE property.

5.5. Threats Outside Scope

Some threats fall beyond the scope of the DY model by construction and hence are addressed separately.

1) Side-channel attacks (timing, power analysis), which is associated with the implementation layer – mitigations are discussed in Discussion section.

2) Denial-of-service attack, where the attacker drops messages (C5) – liveness is harmed but safety properties G1–G4 remains unaffected.

6. Quantum-Resistant Diffie-Hellman Key Exchange Protocol

Here we present the full protocol using algorithmic pseudocode. Alice and Bob wish to establish a shared symmetric key over an untrusted channel under the DY_{QC} threat model defined in Section 5. The protocol is divided in three phases. Figure 2 illustrates the overall flow.

6.1. Setup Phase

Both Alice and Bob independently generate their LWE key pairs and choose their Diffie-Hellman private exponents. They also publicly agree on a large safe prime P and generator g.

Algorithm 1: Setup Phase

Input: Security parameters (n, m, q, η), prime P, generator g

Alice performs:

1. Sample $\mathbf{A}_a \in \mathbb{Z}_q^{m\times n}/(x^n+1)$ uniformly at random
2. Sample $s_a \in \mathbb{Z}_q^{m\times n}/(x^n+1)$ as the secret key vector
3. Sample $e_a \in \mathbb{Z}_q^{m\times n}/(x^n+1)$ from centred binomial distribution $B(\eta)$
4. Compute $\mathbf{b}_a = (\mathbf{A}_a \cdot s_a + e_a) \bmod q$
5. Choose random private exponent $x \in \mathbb{Z}_P$
6. Obtain CA certificate $cert_a$ binding $(\mathbf{A}_a, \mathbf{b}_a, P, g)$ to Alice's identity
7. Public key: $(\mathbf{A}_a, \mathbf{b}_a, P, g)$, Certificate: $cert_a$; Secret key: (s_a, x)

Bob performs (likewise):

1–3. Sample $\mathbf{A}_b, s_b, e_b$ similarly
4. Compute $\mathbf{b}_b = (\mathbf{A}_b \cdot s_b + e_b) \bmod q$
5. Choose random private exponent $y \in \mathbb{Z}_P$
6. Obtain CA certificate $cert_b$ binding $(\mathbf{A}_b, \mathbf{b}_b, P, g)$ to Bob's identity
7. Public key: $(\mathbf{A}_b, \mathbf{b}_b, P, g)$, Certificate: $cert_b$; Secret key: (s_b, y)

Both parties: Exchange and verify each other's CA certificates

6.2. Transmission Phase

Unlike classical DHKE, the modular exponents values are encrypted using LWE before transmission, hiding the DLP parameters from the attacker. Each ciphertext is signed with the sender's Dilithium secret key to ensure authenticity.

Algorithm 2: Transmission Phase

Alice transmits to Bob:

1. Compute $X = g^x \bmod P$
2. Encrypt X as a polynomial: $\mu(X) = (X_0 + X_1 x + \cdots + X_{n-1}x^{n-1})/(x^n+1)$ where X_i is the i-th bit of the binary representation of X
3. Sample $d_b \in \{0,1\}^{\lfloor \log_2 P \rfloor}$ uniformly at random
4. Compute ciphertext: $C_X = d_b \cdot \mathbf{b}_b + \lfloor q/2 \rfloor \mu(X)$
5. Compute Dilithium signature: $\sigma_A = \mathrm{Sign}(sk_{CD_A}, (C_X, d_b))$
6. Send (C_X, d_b, σ_A) to Bob

Bob transmits to Alice (likewise):

1. Compute $Y = g^y \bmod P$
2. Encode Y as polynomial $\mu(Y)$
3. Sample $d_a \in \{0,1\}^{\lfloor \log_2 P \rfloor}$ uniformly at random
4. Compute ciphertext: $C_Y = d_a \cdot \mathbf{b}_a + \lfloor q/2 \rfloor \mu(Y)$
5. Compute Dilithium signature: $\sigma_B = \mathrm{Sign}(sk_{CD_B}, (C_Y, d_a))$
6. Send (C_Y, d_a, σ_B) to Alice

On receipt, each party first verifies the Dilithium signature.

Alice: $\mathrm{Verify}(pk_B, (C_Y, d_a), \sigma_B)$; abort if invalid
Bob: $\mathrm{Verify}(pk_A, (C_X, d_b), \sigma_A)$; abort if invalid

6.3. Shared Key Construction Phase

After verifying the signature and confirming the authenticity, each party decrypts the received ciphertext to eventually compute the shared key.

Algorithm 3: Shared Key Construction Phase

Bob recovers X:

1. Compute $M_X = C_X - d_b \cdot \mathbf{A}_b \cdot s_b$
2. For each bit i: $X_i = 0$ if $(M_X)_i \approx 0$; $X_i = 1$ if $(M_X)_i \approx q/2$
3. Reconstruct integer X from bits X_i
4. Compute shared key: $K_b = X^y \bmod P = (g^x)^y \bmod P = g^{xy} \bmod P$

Alice recovers Y (likewise):

1. Compute $M_Y = C_Y - d_a \cdot \mathbf{A}_a \cdot s_a$
2. For each bit i: $Y_i = 0$ if $(M_Y)_i \approx 0$; $Y_i = 1$ if $(M_Y)_i \approx q/2$
3. Reconstruct integer Y from bits Y_i
4. Compute shared key: $K_a = Y^x \bmod P = (g^y)^x \bmod P = g^{xy} \bmod P$

Result: $K_a = K_b = g^{xy} \bmod P$ (shared symmetric key)

7. Parameter Specification

Proper parameter selection is important for practitioners when deploying this protocol. We follow the NIST post-quantum standardized recommendations (FIPS 203, based on CRYSTALS-Kyber [28], a LWE-KEM) as a reference.

7.1. LWE Parameters

Polynomial Dimension n: Choose $n \in \{512, 768, 1024\}$ correspondingly for equivalent AES {128-bit, 192-bit, 256-bit} level classical security, respectively.

Modulus q: $q = 3329$, as specified in Kyber, enabling efficient NTT-based polynomial multiplication.

Number of Public Equations m: Set $m \approx 2n \log q$ for sufficient security, as per the LWE parameter guidelines [17].

Noise Distribution: Use the centred binomial distribution B_η with $\eta = 3$ for $n = 512$ and $\eta = 2$ for $n = \{768, 1024\}$ according to the standard specification for the CRYSTALS-Kyber [28].

7.2. DHKE Parameters

Prime P: Choose a safe prime of at least 2048 bits. Using standardized groups (e.g., RFC 3526 Group 14 or above) is recommended.

Generator g: Pick a standard generator for the chosen group.

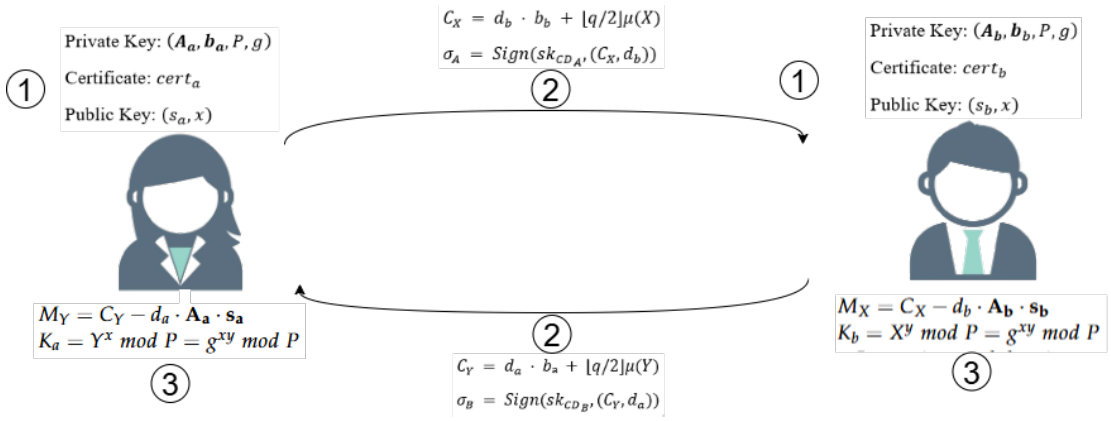

Figure 2. Quantum-Resistant Diffie-Hellman Key Exchange Protocol

7.3. Security-Performance Trade-off

Practitioners should follow the NIST FIPS 203 for up-to-date parameter specifications. Higher security levels comes at a price of high computational overhead due to the costly polynomial matrix multiplication in LWE, introducing latency compared to classical DHKE. It will be more suitable for symmetric key distribution, rather than back-and-forth communication.

8. Complexity Analysis

For complexity analysis, the speculation is done from two perspectives, from the communicating parties end and from the attackers end. As the full protocol is proposed using classical approach, so the first perspective is only discussed according to classical machine's time complexity. But, for the later one, both the classical and quantum threats are analyzed.

8.1. From User's Perspective

8.1.1. Setup Phase

In set up phase, for constructing the public and private keys, both Alice and Bob has to execute some operations. Let, the number of equations revealed publicly for the LWE is m and each equation is of degree n, then we have the random polynomial matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}/(x^n+1)$. The public vector $\mathbf{b} \in \mathbb{Z}_q^m$ is calculated as $\mathbf{b} = \mathbf{A} \cdot \mathbf{s} + \mathbf{e}$. This multiplication takes $\mathcal{O}(mn)$ modular computations. For primality testing for the prime P, the Rabin-Miller algorithm is the standard approach and it takes $\mathcal{O}(k \cdot \log^3 P)$ where k is the number of random bases used for higher accuracy. And for finding the generator g, the average complexity is $\mathcal{O}(\log^4 P)$ in a finite cyclic group [16]. In this step, the higher complexity is $\mathcal{O}(mn)$, so this dictates the total time complexity.

8.1.2. Transmission Phase

Both of the users first execute modular exponentiation and for this, it takes only $\mathcal{O}(\log x)$ steps, where x is the exponent, but this is negligible compared to other complexities.

For using LWE, the standard practice is to set the number of public equations, m and the degree of the equations, n, such that $m \approx 2n \log q$ [17]. If anyone wants to send a single bit, he needs to compute cipher, $C = \mathbf{d} \cdot \mathbf{b} + \lfloor q/2 \rfloor \mu$ where $\mathbf{d} \in \{0,1\}^m$ and $\mu \in \{0,1\}$ is the message bit. So, this step takes only $\mathcal{O}(m)$ executions. Now, both the communicating parties will need $\mathcal{O}(Lm)$ executions for sending an L length string, where $L = \lceil \log P \rceil$.

8.1.3. Shared Key Construction Phase

For revealing the X and Y, the participants compute $C - \mathbf{d} \cdot \mathbf{A} \cdot \mathbf{s}$. Here, the subtraction and multiplication overall takes $\mathcal{O}(mn)$ calculations. To decrypt a string of L bit length using LWE, the complexity will rise to $\mathcal{O}(Lmn) = \mathcal{O}(Ln^2 \log q) = \mathcal{O}(\lceil \log P \rceil n^2 \log q)$ which is the dominant part of this step. They also need to compute modular exponentiation, but that has insignificant complexity.

8.2. From Attacker's Perspective

In this section we analyze the complexity to break the protocol. As there can either be a classical or a quantum threat, we will give light on both type of attacks. This section makes it clear that how promising this protocol can be.

8.2.1. Classical Attacks

Any attacker either can target the DLP part or the LWE part for attacking. But in the proposed protocol, the LWE creates an extra protective layer for the DLP. So, the attacker will tend to break the LWE.

One possible attack on LWE is the lattice reduction technique using LLL algorithm [18]. It has polynomial time complexity but with weak reduction quality. Another approach can be taken using the BKZ algorithm [19] with a stronger lattice reduction method but with an extra overhead with an exponential time complexity on the block size [20]. Other classical attacks on LWE has practically exponential time complexity which makes our proposal resistant against any classical threats.

Now, if we think about the DLP's security, using the generic algorithms like, the baby step giant step or the Pollard's Rho algorithm, it takes $\mathcal{O}(\sqrt{P})$ for group order P where P is a very large prime [21]. Other approaches have sub-exponential time complexity. Hence, classically it is infeasible to break the DLP.

8.2.2. Quantum Attacks

Alike the classical approach, quantum attack can either be posed on the LWE part or the DLP part, but the attack

is likely to pose threat on LWE first as LWE keeps the DLP protected.

Currently there are several candidates approaches, but none of them are efficient enough to break LWE with standard cryptographic parameters.

Kuperberg's algorithm achieves sub-exponential complexity for the dihedral hidden subgroup problem, but fails to break generic LWE with standard parameters [22]. In another research, the authors proposed quantum-classical hybrid algorithms with an Ising model that encode LWE problems into an Ising Hamiltonian [23]. This approach shows potential for solving LWE with small parameters but do not scale to break practical cryptographic instances. Also, the quantum holographic attacks on LWE requires exponential time [24]. Shor's algorithm remains inapplicable to LWE as the noises break the periodic structure for period finding attack. Hence, the LWE part remains unthreatened by any quantum attacks.

The DLP part can be broken by Shor's algorithm in polynomial time [2]. But as the LWE protects the DLP parameters from being exposed, the DLP will remain unthreatened as long as the LWE does not fall.

9. Security Analysis and Discussion

9.1. MITM Resistance

Under the DY_{QC} model, the adversary I cannot satisfy the assertion $\mathrm{ver}(\sigma, \langle C_X, n_a, \mathrm{Bob}\rangle, vk_{CD}(\mathrm{Alice})) = \text{true}$ without holding $sk_{CD}(\mathrm{Alice})$. Any injected or modified ciphertext fails signature verification, and hence the protocol aborts (Goal G2, Section 5.4). Also, the LWE encryption hides the DH value $X = g^x \bmod P$ from the adversary (Goal G1), so even a quantum-capable I applying Shor's algorithm cannot recover x without first breaking LWE — which is computationally infeasible under our parameters. This resolves the MITM vulnerability inherent in classical DHKE.

9.2. Forward Secrecy

Ephemeral exponents x and y ensure that compromise of long-term keys (LWE secret keys or Dilithium signing keys) does not expose past session keys. This is a key advantage inherited from DHKE over simple LWE key encapsulation with certificate-bound long-term keys.

9.3. Single Point of Failure Consideration

We acknowledge that the security of the DLP part is dependent on the LWE remaining unbroken. This is indeed a sequential security dependency, not a parallel one.

However, LWE is currently considered one of the most thoroughly studied post-quantum hard problems that gives base of the NIST's finalized post-quantum cryptographic standards (CRYSTALS-Dilithium, CRYSTALS-Kyber). If any approach breaks LWE, that would lead to a fundamental breakthrough in the post-quantum cryptography field, affecting all other protocols – not just our one. For future work, we suggest exploring a design where the LWE and DLP components are combined in parallel so that both must be broken independently to compromise the protocol.

9.4. Side Channel Attack Consideration

Implementation threats can be more dangerous sometimes than theoretical ones in practice. We recommend the following countermeasures during practical deployment:

Timing attacks: All modular exponentiation and LWE operations must be implemented in constant time. Montgomery ladder or Fixed-window methods are suggested for modular exponentiation.

Power Analysis: Masked implementation should be used to defend against Simple Power Analysis (SPA) and Differential Power Analysis (DPA).

Noise generation: Rejection sampling should be used while generating the error vector from a binomial distribution to prevent timing leakage.

10. Result

Table 1 provides a comparative analysis of the proposed protocol against related approaches. The classical DHKE protocol is prone to MITM attacks and quantum threats. Our proposal is the only approach that simultaneously provides authentication, MITM protection, quantum resistance, and requires no quantum hardware.

11. Conclusion

Due to the inherent issues in the DHKE protocol, different improvements have been made over time. Nevertheless, these approaches have their own drawbacks. Our approach solves the threats to DHKE and outperforms other works. This work paves a new way for sharing symmetric keys using forward secrecy, ensuring the authenticity of the parties. This protocol is robust against any classical or quantum threats, hence a promising candidate for a post-quantum key distribution protocol.

We have formalised the threat model using the Dolev-Yao symbolic attacker model extended to quantum-capable adversaries (DY_{QC}), and formal proofs of four security goals: secrecy, mutual authentication, replay resistance, and forward secrecy. The protocol was also presented in standard pseudocode notation, concrete parameter selection guidance was provided aligned with NIST post-quantum standards, and side-channel attack mitigations were discussed. The protocol requires no quantum hardware and is a feasible upgrade solution to post-quantum symmetric key distribution from existing classical DH-dependent systems.

Future work includes: formal symbolic verification of the protocol (e.g., using ProVerif or Tamarin); exploration of a parallel-security composition to eliminate the sequential LWE-DLP dependency; and implementation benchmarking on resource-constrained devices.

References

[1] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. IT-22, Nov. 1976.

[2] M. Roetteler et al., "Quantum resource estimates for computing elliptic curve discrete logarithms," *arXiv preprint arXiv:1706.06752*, 2017.

Table 1. Comparison with other approaches

Papers	Authentication	MITM Attack	Pre-shared Secret Key	Quantum Resource	Classical Threat	Quantum Threat
DHKE [1]	No	Yes	Not Needed	Not Needed	No	Yes
Tefra et al. [12]	Yes	Yes	Needed	Not Needed	No	Yes
Rakel et al. [13]	No	Yes	Not Needed	Not Needed	No	Yes
Mahmood et al. [3]	Yes	Yes	Not Needed	Not Needed	No	Yes
Georgios et al. [14]	No	Yes	Not Needed	Needed	No	No
Dima et al. [15]	No	Yes	Not Needed	Not Needed	No	Yes
Our Proposal	Yes	Yes	Not Needed	Not Needed	No	No

[3] M. K. Ibrahem, "Modification of Diffie-Hellman key exchange algorithm for zero knowledge proof," in *2012 International Conference on Future Communication Networks*, pp. 147–152, 2012. https://doi.org/10.1109/ICFCN.2012.6206859.

[4] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Review*, vol. 41, no. 2, pp. 303–332, 1999. https://doi.org/10.1137/S0036144598347011.

[5] National Institute of Standards and Technology, "NIST releases first 3 finalized post-quantum encryption standards," August 2024. https://www.nist.gov/news-events/news/2024/08/nist-releases-first-3-finalized-post-quantum-encryption-standards.

[6] S. Khot, "Hardness of approximating the shortest vector problem in lattices," *Journal of the ACM*, vol. 52, no. 5, pp. 789–808, 2005. https://doi.org/10.1145/1089023.1089027.

[7] H. Bennett, A. Golovnev, and N. Stephens-Davidowitz, "On the quantitative hardness of CVP," *arXiv preprint arXiv:1704.03928*, 2017.

[8] O. Regev, "The learning with errors problem (invited survey)," in *Proceedings of the 25th Annual IEEE Conference on Computational Complexity*, pp. 191–204, 2010. https://doi.org/10.1109/CCC.2010.26.

[9] Z. Brakerski et al., "Classical hardness of learning with errors," *arXiv preprint arXiv:1306.0281*, 2013.

[10] Z. Zhao and J. Ding, "Practical improvements on BKZ algorithm," in Dolev, Gudes, and Paillier (Eds.), pp. 273–284, 2023.

[11] Z. Zeng et al., "Quantum-classical hybrid algorithm for solving the learning-with-errors problem on NISQ devices," *Communications Physics*, vol. 8, no. 1, p. 291, 2025. https://doi.org/10.1038/s42005-025-02126-w.

[12] T. Galu, A. Adeyelu, and S. Otor, "An improved Diffie-Hellman scheme for mitigating an eavesdropping attack on a network," *International Journal of Innovative Science and Research Technology (IJISRT)*, pp. 3101–3108, June 2024. https://doi.org/10.38124/ijisrt/IJISRT24APR2479.

[13] R. Haakegaard and J. Lang, "The elliptic curve Diffie-Hellman (ECDH)," 2015. [Online]. Available: https://koclab.cs.ucsb.edu/teaching/ecc/project/2015Projects/Haakegaard+Lang.pdf.

[14] G. M. Nikolopoulos, "Quantum Diffie–Hellman key exchange," *APL Quantum*, vol. 2, 2025. https://doi.org/10.1063/5.0242473.

[15] D. Alrwashdeh et al., "On two novel generalized versions of Diffie-Hellman key exchange algorithm based on neutrosophic and split-complex integers and their complexity analysis," *International Journal of Neutrosophic Science*, vol. 25, no. 2, pp. 01–10, 2025.

[16] T. Adamski and W. Nowakowski, "The average time complexity of probabilistic algorithms for finding generators in finite cyclic groups," *Bulletin of the Polish Academy of Sciences. Technical Sciences*, vol. 63, no. 4, pp. 989–996, 2015.

[17] H. Corrigan-Gibbs and Y. Kalai, "Public-key encryption from LWE & implementing lattice-based cryptosystems," MIT 6.5610 Lecture Notes, Lecture 8, February 26, 2024. https://65610.csail.mit.edu/2024/lec/l07-pke.pdf.

[18] A. K. Lenstra, H. W. Lenstra, and L. Lovász, "Factoring polynomials with rational coefficients," *Mathematische Annalen*, vol. 261, pp. 515–534, 1982. https://doi.org/10.1007/BF01457454.

[19] C.-P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Mathematical Programming*, vol. 66, no. 1, pp. 181–199, 1994. https://doi.org/10.1007/BF01581144.

[20] T. Köppl et al., "A parameter study for LLL and BKZ with application to shortest vector problems," *arXiv preprint arXiv:2502.05160*, 2025.

[21] G. S. Sia, "A brief survey of the discrete logarithm problem," 2011.

[22] G. Kuperberg, "A subexponential-time quantum algorithm for the dihedral hidden subgroup problem," *SIAM Journal on Computing*, vol. 35, no. 1, pp. 170–188, 2005. https://doi.org/10.1137/S0097539703436345.

[23] M. Zheng et al., "Quantum-classical hybrid algorithm for solving the learning-with-errors problem on NISQ devices," *Communications Physics*, vol. 8, no. 1, p. 291, 2025. https://doi.org/10.1038/s42005-025-02126-w.

[24] Y. Wang, X. Jin, and J. Liu, "Learning with errors may remain hard against quantum holographic attacks," *arXiv preprint arXiv:2509.22833*, 2025.

[25] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983. https://doi.org/10.1109/TIT.1983.1056650.

[26] V. Lyubashevsky et al., "CRYSTALS-Dilithium: A lattice-based digital signature scheme," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2018, no. 1, pp. 238–268, 2018. https://doi.org/10.13154/tches.v2018.i1.238-268.

[27] National Institute of Standards and Technology, "Recommendations for key-establishment schemes," NIST SP 800-227 (Initial Public Draft), 2024.

[28] J. Bos et al., "CRYSTALS-Kyber: A CCA-secure module-lattice-based KEM," Cryptology ePrint Archive, Paper 2017/634, 2017.