Efficient Post-Quantum TLS Handshakes using Identity-Based Key Exchange from Lattices

Efficient Post-Quantum TLS Handshakes using Identity-Based Key Exchange from Lattices

Utsav Banerjee and Anantha P. Chandrakasan
Dept. of EECS, Massachusetts Institute of Technology, Cambridge, MA, USA

***Abstract*—Identity-Based Encryption (IBE) is considered an alternative to traditional certificate-based public key cryptography to reduce communication overheads in wireless sensor networks. In this work, we build on the well-known lattice-based DLP-IBE scheme to construct an ID-based certificate-less authenticated key exchange for post-quantum Transport Layer Security (TLS) handshakes. We also propose concrete parameters for the underlying lattice computations and provide detailed implementation results. Finally, we compare the combined computation and communication cost of our ID-based certificate-less handshake with the traditional certificate-based handshake, both using lattice-based algorithms at similar post-quantum security levels, and show that our ID-based handshake is $3.7\times$ more energy-efficient, thus highlighting the advantage of ID-based key exchange for post-quantum TLS.**

I. Introduction

Wireless sensor networks (WSNs) consist of electronic devices connected together and exchanging confidential data, and public key cryptography (PKC) is widely used to secure these communication channels. Traditional PKC uses key exchange, digital signatures and digital certificates to perform mutual authentication and generate shared encryption keys. However, using certificates poses significant storage and communication overheads [1]. While it is possible to cache certificates locally if each sensor node communicates only with a fixed small set of other nodes, this method quickly becomes impractical as the network grows larger and more nodes talk to each other. Furthermore, addition of new nodes in the network requires updating such local certificate caches, which can be a problem in wireless ad-hoc networks where nodes are allowed to join or leave the network on-the-fly. Identity-based encryption (IBE) has been proposed as a potential solution to such problems [1]. IBE uses unique digital identities (such as IP addresses) of sensor nodes to perform public key cryptography, thus avoiding the use of certificates altogether and reducing communication overheads. The most well-known IBE construction is based on bilinear pairings from elliptic curves [24]. However, pairing computations are an order of magnitude more expensive than traditional elliptic curve cryptography (ECC) [27], which makes the benefits of using pairing-based IBE only marginal.

With the advent of quantum computing, new public key cryptography algorithms are being developed which are secure against quantum attacks [2], and lattice-based cryptography has emerged as a prime candidate [3]. The DLP-IBE scheme [9] is the most efficient lattice-based IBE construction till date. In this work, we build on this IBE scheme to construct quantum-secure certificate-less authenticated key exchange which is integrated with the Transport Layer Security (TLS) protocol [16] to save communication costs by eliminating the need to exchange certificates. This is the first demonstration of post-quantum ID-based certificate-less TLS handshake. We also propose concrete parameters for the IBE scheme, and report measured performance as implemented on a custom chip with hardware accelerator for lattice cryptography [14]. We compare the combined computation and communication cost of our ID-based handshake with the traditional certificate-based handshake, both using lattice-based algorithms at similar security levels, and show that our ID-based handshake is $3.7\times$ more energy-efficient, thus demonstrating the superiority of ID-based TLS in the post-quantum scenario.

II. Background

In this section, we introduce the mathematical notation used in this paper and provide a brief overview of lattice-based cryptography and identity-based encryption (IBE).

A. Lattice-based Cryptography using Ring-LWE

Throughout this paper, we will work over the polynomial ring modulo $(x^n + 1)$ of integers modulo prime q, denoted as $\mathcal{R}_q = \mathbb{Z}_q[x]/(x^n + 1)$, where n is a power of 2 and $q \equiv 1 \mod 2n$ to allow fast polynomial multiplications in $\mathcal{R}_q$ using the number theoretic transform (NTT) [4], [5]. Polynomials in $\mathcal{R}_q$ are written using lower-case symbols, $\|$ denotes concatenation, $\star$ denotes polynomial multiplication and $\lfloor \cdot \rceil$ denotes coefficient-wise rounding of polynomials. All symmetric cryptography functions are instantiated using the NIST standard algorithms AES [6] and SHA3 [7].

The ID-based schemes described in this paper are based on the Ring-LWE problem [3] which states that given $(a, a \star s + e)$, it is difficult to determine secret polynomial $s \in \mathcal{R}_q$, where polynomial $a \in \mathcal{R}_q$ is sampled uniformly at random and the coefficients of error polynomial e are small samples from an error distribution χ. Further details about Ring-LWE crypto-systems are available in [3].

B. Overview of Identity-Based Encryption

Identity-Based Encryption (IBE) is a type of public-key encryption where public keys of users are derived from their identities, e.g., e-mail, IP addresses, etc. Unlike traditional protocols where user public keys are obtained from certificates, IBE has the unique advantage of not requiring certificate storage and verification. A trusted third party, known as

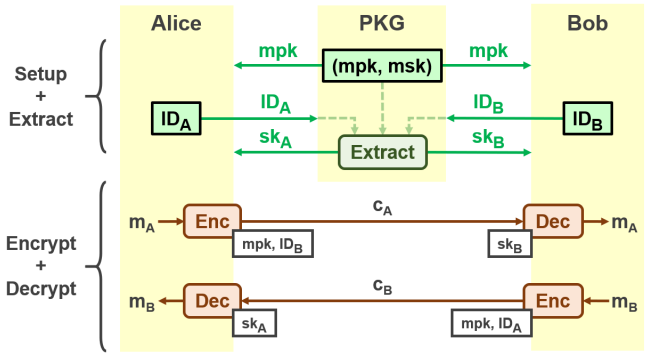

Fig. 1. Summary of steps in ID-based encryption scheme.

Private Key Generator (PKG), is required to generate user keys, analogous to Certificate Authority (CA) in the traditional setting. Given security parameter λ, an IBE scheme consists of the following four probabilistic polynomial time algorithms:

- Setup $(1^\lambda) \rightarrow (mpk,\ msk)$: used to generate master public key mpk and master secret key msk of the PKG.
- Extract $(mpk,\ msk,\ ID) \rightarrow sk_{ID}$: used by the PKG to generate secret key sk_{ID} of an user with identity ID.
- Encrypt $(mpk,\ ID,\ m) \rightarrow c$: sender encrypts message m using mpk and receiver's public key derived from their identity ID, and outputs ciphertext c.
- Decrypt $(sk_{ID},\ c) \rightarrow \{m, \perp\}$: receiver decrypts ciphertext c using their secret key sk_{ID}, and outputs either message m or $\perp$ if the ciphertext is invalid.

These algorithms are summarized in Fig. 1. The IBE scheme is correct if, for any message m and identity ID, the following equality holds with overwhelming probability:

Decrypt $(sk_{ID},$ Encrypt $(mpk,\ ID,\ m)) = m$

The Setup and Extract steps are performed very infrequently. Once the keys are set up and stored, the Encrypt and Decrypt steps are used for ID-based encryption and decryption.

III. Lattice-Based IBE and Implementation

The first lattice-based IBE crypto-system was proposed by Gentry et al. [8], but had ciphertexts of the order of millions of bits, thus making it impractical. Several improvements have been proposed over the past years, and the most efficient construction till date is the DLP-IBE scheme [9] which uses NTRU lattices for key generation and Ring-LWE for encryption to achieve public keys of size $O(n)$ and ciphertexts of size $O(2n)$, where n is the degree of polynomial ring $\mathcal{R}_q$.

A. Original CPA-Secure IBE Scheme

The Ring-LWE-based Encrypt and Decrypt functions of the DLP-IBE scheme are described in Algorithms 1 and 2. Details of the Setup and Extract algorithms are available in [9], and we exclude any discussion on them since only the Encrypt and Decrypt algorithms are expected to be executed by constrained embedded devices such as low-power wireless sensor nodes.

Algorithm 1 IND-CPA-Secure ID-based Encryption [9]

function IBE-CPA-Encrypt (mpk, ID, m)
  $r, e_1, e_2 \xleftarrow{\$} \{-1, 0, 1\}^n$; $k \xleftarrow{\$} \{0, 1\}^n$ (uniform)
  $u \leftarrow r \star mpk + e_1 \in \mathcal{R}_q$
  $v \leftarrow r \star H(ID) + e_2 + \lfloor q/2 \rfloor \cdot k \in \mathcal{R}_q$
  $v \leftarrow \lfloor v/2^l \rceil$
  return $(u, v, c = m \oplus H'(k))$

Algorithm 2 IND-CPA-Secure ID-based Decryption [9]

function IBE-CPA-Decrypt $(sk_{ID}, (u, v, c))$
  $v \leftarrow 2^l \cdot v$
  $w \leftarrow v - u \star sk_{ID} \in \mathcal{R}_q$
  $k \leftarrow \lfloor \frac{w}{q/2} \rceil$
  return $m = c \oplus H'(k)$

In the Encrypt step, coefficients of the error polynomials r, e_1 and e_2 are sampled from a discrete probability distribution with support $\{-1, 0, 1\}$, and the coefficients of polynomial k are sampled uniformly from $\{0, 1\}$. The distribution parameters directly affect security and efficiency of the IBE scheme, and we describe our parameter selection in detail in Section III-C, along with the choice of n and q. H is a hash function which maps an arbitrary-length identity string ID to a polynomial in $\mathcal{R}_q$, and H' is another hash function which converts $k \in \mathcal{R}_q$ to a one-time pad of length $mlen$ (equal to the length of message m). The polynomial v is compressed by dropping l least significant bits of each of its coefficients. This causes negligible increase in decryption failure probability as long as $l \leq \lfloor \log_2 q \rfloor - 3$, according to [9].

To verify that the decryption works correctly (with an infinitesimally small probability of failure), we note:

$$\begin{aligned} w &\approx r \star H(ID) + e_2 + \lfloor q/2 \rfloor \cdot k - (r \star mpk + e_1) \star sk_{ID} \\ &= r \star \{H(ID) - mpk \star sk_{ID}\} + e_2 - e_1 \star sk_{ID} + \lfloor q/2 \rfloor \cdot k \\ &= r \star s + e_2 - e_1 \star sk_{ID} + \lfloor q/2 \rfloor \cdot k \end{aligned}$$

since the master public key and user secret key satisfy the property: $mpk \star sk_{ID} + s = H(ID)$, where s is a short element in $\mathcal{R}_q$ [9]. Decryption is correct as long as all coefficients of $r \star s + e_2 - e_1 \star sk_{ID}$ lie in the range $(-q/4, q/4)$.

B. Proposed CCA-Secure IBE Scheme

The original DLP-IBE scheme is only IND-CPA-secure, that is, *indistinguishable under chosen plaintext attacks*, so the same key-pair cannot be used for multiple encryptions. This is not only a problem from a security perspective, but also makes it inefficient because the Setup and Extract steps need to be repeated every time an ID-based encryption is performed. Here, we describe how to make this scheme IND-CCA2-secure, that is, *indistinguishable under adaptive chosen ciphertext attacks*, using the standard Fujisaki-Okamoto transform [12]. The IND-CCA2-secure scheme allows key reuse so that keys can be cached long-term in the sensor nodes.

The key generation phase remains unchanged, and our proposed IND-CCA2-secure IBE scheme is described in

Algorithm 3 IND-CCA2-Secure ID-based Encryption

function IBE-CCA-ENCRYPT (mpk, ID, m)
  $k \xleftarrow{\$} \{0,1\}^n$ (uniform)
  $r \leftarrow F(k \,\|\, \texttt{0x00}) \in \{-1,0,1\}^n$
  $e_1 \leftarrow F(k \,\|\, \texttt{0x01}) \in \{-1,0,1\}^n$
  $e_2 \leftarrow F(k \,\|\, \texttt{0x02}) \in \{-1,0,1\}^n$
  $u \leftarrow r \star mpk + e_1 \in \mathcal{R}_q$
  $v \leftarrow r \star H(ID) + e_2 + \lfloor q/2 \rfloor \cdot k \in \mathcal{R}_q$
  $v \leftarrow \lfloor v/2^l \rceil$
  return $(u, v, c = m \oplus H'(k), d = G(k))$

Algorithm 4 IND-CCA2-Secure ID-based Decryption

function IBE-CCA-DECRYPT $(sk_{ID}, (u, v, c, d))$
  $v \leftarrow 2^l \cdot v$
  $w \leftarrow v - u \star sk_{ID} \in \mathcal{R}_q$
  $k' \leftarrow \lfloor \frac{w}{q/2} \rceil$
  $r' \leftarrow F(k' \,\|\, \texttt{0x00}) \in \{-1,0,1\}^n$
  $e'_1 \leftarrow F(k' \,\|\, \texttt{0x01}) \in \{-1,0,1\}^n$
  $e'_2 \leftarrow F(k' \,\|\, \texttt{0x02}) \in \{-1,0,1\}^n$
  $u' \leftarrow r' \star mpk + e'_1 \in \mathcal{R}_q$
  $v' \leftarrow r' \star H(ID) + e'_2 + \lfloor q/2 \rfloor \cdot k' \in \mathcal{R}_q$
  $v' \leftarrow \lfloor v'/2^l \rceil$
  if $d = G(k')$ **and** $(u, v) = (u', v')$ **then**
    return $m = c \oplus H'(k')$
  else
    return $\perp$

Algorithms 3 and 4. The CCA-secure encryption deterministically derives the error polynomials r, e_1 and e_2 from k instead of sampling them randomly like its CPA-secure counterpart. The CCA-secure decryption actually performs decryption followed by re-encryption to verify that the correct inputs were provided, otherwise the algorithm aborts. Here, F is a hash function which generates error polynomials from k, and G is another hash function which computes a $hlen$-bit digest of the polynomial k. Proof of IND-CCA2 security in the random oracle model follows from [12].

C. Selection of Efficient and Secure Parameters

Unlike classical public key cryptography, Ring-LWE parameter selection is a complex task because of the multitude of parameters involved and their varying effects on security, efficiency and correctness of the encryption scheme. Concrete parameters for the DLP-IBE scheme were proposed in [9], [10], [11] for 80-bit and 192-bit security. In this work, we target 128-bit security level, where $n = 1024$ and $q \approx 2^{23}$, as recommended in [9] and [10]. To ensure that prime q allows efficient modular multiplication, we choose $q = 8380417 = 2^{23} - 2^{13} + 1$ which supports fast Barrett reduction due to its special structure [14]. Also, $q \equiv 1 \bmod 2n$, thus allowing fast polynomial multiplication using NTT. We explore two options for choosing the error probability distribution $\Pr[x]$ for $x \in \{-1, 0, 1\}$: (1) uniform distribution with $\Pr[x = -1] = \Pr[x = 0] = \Pr[x = 1] = 1/3$, and (2) trinary distribution with $\Pr[x = -1] = \Pr[x = 1] = \rho/2$ and $\Pr[x = 0] = 1 - \rho$ for $\rho \in \{1/2, 1/4, 1/8, \cdots\}$. We use the methodology proposed in [15] to analyze security of the IBE scheme for different error distributions with varying standard deviation (σ). In Table I, we show the security levels (in bits) provided by these distributions for our parameters $(n, q) = (1024, 8380417)$. Clearly, the uniform distribution provides highest security, while security provided by the trinary distribution decreases with smaller ρ. Since sampling of error polynomials accounts for bulk of the computation cost of Ring-LWE [14], we also analyze the number of pseudo-random bits required to generate samples from these distributions as an indicator of their efficiency. For sampling a polynomial coefficient from distribution (1), we need to generate 2 uniformly random bits and use rejection sampling, that is, output -1, 0 and 1 when these bits are 00_2, 01_2 and 10_2 respectively, and reject (and repeat the process with 2 more random bits) when they are 11_2. Then, the expected number of random bits to sample uniformly in $\{-1, 0, 1\}$ is

$$= 2 \cdot \tfrac{3}{4} + 4 \cdot \tfrac{1}{4} \cdot \tfrac{3}{4} + 6 \cdot (\tfrac{1}{4})^2 \cdot \tfrac{3}{4} + 8 \cdot (\tfrac{1}{4})^3 \cdot \tfrac{3}{4} + 10 \cdot (\tfrac{1}{4})^4 \cdot \tfrac{3}{4} + \cdots$$
$$= 2 \cdot \tfrac{3}{4} \cdot \{ \textstyle\sum_{i=1}^{\infty} i \cdot (\tfrac{1}{4})^{i-1} \} = \tfrac{3}{2} \cdot \tfrac{1}{(1-\frac{1}{4})^2} = \tfrac{8}{3}$$

and the total number of random bits required for sampling n such polynomial coefficients is $8n/3$ on average. For sampling a polynomial coefficient from distribution (2) where $1/\rho$ is a power of two, we need to generate $\log_2(2/\rho)$ uniformly random bits and then output -1 when these bits are all zeros, 1 when they are all ones, and 0 otherwise. Rejection sampling is not necessary in this case, and sampling n such polynomial coefficients always requires $n \log_2(2/\rho)$ random bits. We choose the trinary distribution with $\rho = 1/2$ because it requires the smallest number of random bits, as shown in Table I. There is slight reduction in security of the IBE scheme compared to using the uniform distribution, but it still remains well above our target 128-bit security level.

TABLE I
SECURITY OF IBE SCHEME WITH DIFFERENT ERROR DISTRIBUTIONS FOR PROPOSED PARAMETERS $(n, q) = (1024, 8380417)$

Distribution	ρ	σ	Security Level	Random Bits
Uniform	-	$\sqrt{2/3}$	143	≈ 2731
Trinary	1/2	$1/\sqrt{2}$	141	2048
	1/4	$1/2$	134	3072
	1/8	$1/2\sqrt{2}$	129	4096

Finally, we summarize the sizes of the master public key and the ciphertext for both CPA-secure and CCA-secure IBE schemes with our proposed parameters:

IBE Scheme	Public Key Size (bytes)	Ciphertext Size (bytes)
IND-CPA-Secure	2,944	3,712
IND-CCA2-Secure	2,944	3,744

where the ciphertext compression parameter is set to $l = 18$, similar to [10]. The size of the public key is $n \lceil \log_2 q \rceil$ bits, while the ciphertext is $n\,(2 \lceil \log_2 q \rceil - l) + mlen$ and

TABLE II
PERFORMANCE AND ENERGY CONSUMPTION OF IBE IMPLEMENTATION

IBE Scheme	Encrypt		Decrypt	
	Cycles	μJ	Cycles	μJ
IND-CPA-Secure	95,369	10.15	111,652	11.91
IND-CCA2-Secure	106,980	11.45	194,171	20.75

$n\,(2\,\lceil \log_2 q \rceil - l) + mlen + hlen$ bits long for the CPA-secure and CCA-secure IBE schemes respectively, with $mlen = 1024$ bits and $hlen = 256$ bits.

D. Implementation Results

We implement the IBE scheme on a custom chip [13], [14] we have designed to accelerate lattice-based cryptography. It consists of a 32-bit RISC-V micro-processor (Dhrystone performance comparable to ARM Cortex-M0) with a programmable lattice-crypto accelerator which supports configurable parameters (n, q), choice of several error distributions with flexible standard deviations and uses a fast SHA-3 core for pseudo-random number generation and hashing.

For our NTT implementation, we choose the n-th and $2n$-th roots of unity modulo q to be $\omega = 10730$ and $\psi = 1306$ respectively. We instantiate the hash functions $H : \{0,1\}^* \rightarrow \mathcal{R}_q$, $H' : \mathcal{R}_q \rightarrow \{0,1\}^{mlen}$ and $F : \mathcal{R}_q \times \{0,1\}^8 \rightarrow \mathcal{R}_q$ using the SHA-3-based extendable output function SHAKE-256, and $G : \mathcal{R}_q \rightarrow \{0,1\}^{hlen}$ using SHA3-256. The cycle counts and energy consumption of ID-based encryption decryption, both CPA-secure and CCA-secure, are reported in Table II as measured on our chip operating at 1.1 V and 72 MHz. Our hardware-accelerated CCA-secure ID-based encryption and decryption take 1.5 ms and 2.7 ms respectively, which are fast enough for practical applications. Also, our implementation is constant-time and secure against timing and simple power analysis side-channel attacks [14].

IV. IDENTITY-BASED KEY EXCHANGE FOR TLS

The Transport Layer Security (TLS) protocol [16] is widely used to provide end-to-end network security for internet communications. It guarantees the three most important security attributes – authentication, confidentiality and integrity of the communications channel, even in the presence of malicious network infrastructure. TLS 1.2 is currently the most used version of TLS, and TLS 1.3 has recently been standardized with several improvements over its predecessor [16].

Fig. 2 shows the TLS 1.3 handshake with certificate-based mutually authenticated key exchange. The *ClientHello* and *ServerHello* messages contain respective shares of the key exchange, while the *CertificateVerify* messages contain signatures over the handshake transcript used for authentication. Each *Certificate* message contains the respective party's public key signed by the certificate authority (CA) (assuming a single-level certification hierarchy). The CA public key, known to both parties, is used to verify these certificates. The public keys in these certificates are then used to verify the *CertificateVerify* signatures. Table III shows the key share,

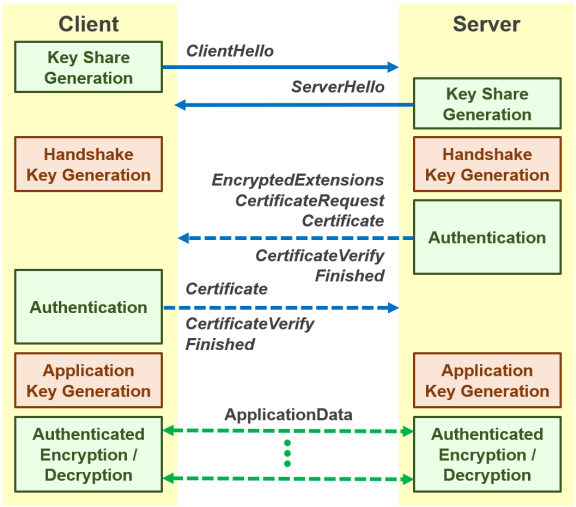

Fig. 2. The TLS 1.3 handshake with mutual authentication and key exchange (blue and green arrows show handshake messages and application data respectively; dashed arrows indicate that encrypted communication).

TABLE III
KEY SHARE, PUBLIC KEY AND SIGNATURE SIZES FOR TLS HANDSHAKE

	Pre-Quantum	Post-Quantum
Client Key Share Size (bytes)	64	928
Server Key Share Size (bytes)	64	1,088
Cert. Public Key Size (bytes)	64	14,880
Signature Size (bytes)	64	2,592

certificate public key and signature sizes for a standard *pre-quantum* TLS handshake which uses elliptic curve cryptography [17]. We assume that the NIST P-256 curve is used for both Elliptic Curve Diffie-Hellman Key Exchange (ECDHE) and Elliptic Curve Digital Signature Algorithm (ECDSA).

There have been some recent efforts in implementing post-quantum TLS [18], [19], [20], [21] and post-quantum certificates [22]. We focus on lattice-based cryptography not only due to its computational efficiency but also because it is the only family of post-quantum public key cryptography algorithms offering efficient ID-based encryption, key encapsulation and signature schemes. We use Ring-LWE for all the public key cryptography in our implementation to maintain the same notion of security, given DLP-IBE also uses Ring-LWE for encryption and decryption. We consider Ring-LWE-based NewHope-512 [4] and qTesla-I [5] (security level similar to our selected parameters for the IBE scheme) as the key encapsulation and signature schemes respectively for *post-quantum* TLS handshake. The corresponding key sizes are shown in Table III.

We refer to [17] for typical TLS message sizes and calculate the total communication costs for certificate-based pre-quantum and post-quantum TLS handshake as 1,820 bytes and 43,452 bytes respectively, that is, post-quantum TLS is $24\times$ more expensive. This is the motivation for our ID-based certificate-less authenticated key exchange for post-quantum

Algorithm 5 IND-CCA2-Secure ID-based Encapsulation

function ID-KEM-CCA-ENCAPS (mpk, ID)
- $k \xleftarrow{\$} \{0,1\}^n$ (uniform)
- $r \leftarrow F(k \,\|\, \texttt{0x00}) \in \{-1,0,1\}^n$
- $e_1 \leftarrow F(k \,\|\, \texttt{0x01}) \in \{-1,0,1\}^n$
- $e_2 \leftarrow F(k \,\|\, \texttt{0x02}) \in \{-1,0,1\}^n$
- $u \leftarrow r \star mpk + e_1 \in \mathcal{R}_q$
- $v \leftarrow r \star H(ID) + e_2 + \lfloor q/2 \rfloor \cdot k \in \mathcal{R}_q$
- $v \leftarrow \lfloor v/2^l \rceil$
- $c \leftarrow H'(k)$
- **return** $\mathbb{K} = G'(u \,\|\, v \,\|\, c \,\|\, k)$, (u, v, c)

Algorithm 6 IND-CCA2-Secure ID-based Decapsulation

function ID-KEM-CCA-DECAPS $(sk_{ID}, s, (u, v, c))$
- $v \leftarrow 2^l \cdot v$
- $w \leftarrow v - u \star sk_{ID} \in \mathcal{R}_q$
- $k' \leftarrow \lfloor \frac{w}{q/2} \rceil$
- $r' \leftarrow F(k' \,\|\, \texttt{0x00}) \in \{-1,0,1\}^n$
- $e'_1 \leftarrow F(k' \,\|\, \texttt{0x01}) \in \{-1,0,1\}^n$
- $e'_2 \leftarrow F(k' \,\|\, \texttt{0x02}) \in \{-1,0,1\}^n$
- $u' \leftarrow r' \star mpk + e'_1 \in \mathcal{R}_q$
- $v' \leftarrow r' \star H(ID) + e'_2 + \lfloor q/2 \rfloor \cdot k' \in \mathcal{R}_q$
- $v' \leftarrow \lfloor v'/2^l \rceil$
- **if** $(u, v, c) = (u', v', H'(k'))$ **then**
  - **return** $\mathbb{K} = G'(u \,\|\, v \,\|\, c \,\|\, k')$
- **else**
  - **return** $\mathbb{K} = G'(u \,\|\, v \,\|\, c \,\|\, s)$

TLS, where each party stores only the master public key and certificates need not be exchanged. While ID-based TLS was proposed long ago in [23], where the core IBE scheme was based on bilinear pairings from elliptic curves, it was not particularly beneficial since keys were already small. Next, we describe our lattice-based construction of ID-based authenticated key exchange and show that ID-based TLS is a great candidate in the post-quantum scenario where signatures and public keys are significantly larger.

First, we convert the CCA-secure IBE scheme from Section III-B into a CCA-secure ID-based key encapsulation mechanism (KEM), based on the generic constructions from [24]. Key encapsulation consists of the following algorithms:

- KeyGen $(1^\lambda) \rightarrow (pk,\ sk)$: used to generate public key pk and secret key sk.
- Encaps $(pk) \rightarrow (\mathbb{K},\ c)$: encapsulates shared secret $\mathbb{K}$ into ciphertext c using public key pk.
- Decaps $(sk,\ c) \rightarrow \mathbb{K}$: decapsulates ciphertext c into shared secret $\mathbb{K}$ using secret key sk.

For ID-based KEM, the key generation step comprises the Setup and Extract algorithms described in Section III, along with a secret polynomial s sampled uniformly from $\{0,1\}^n$. The ID-based Encaps and Decaps steps are shown in Algorithms 5 and 6 respectively. In case of decryption failure, $\mathbb{K}$ is generated using s instead of k' in the decapsulation algorithm. Size of the ciphertext c is 3,712 bytes, the shared secret $\mathbb{K}$ is 256 bits long and the hash function $G' : \{0,1\}^* \rightarrow \{0,1\}^{256}$ is instantiated using SHA3-256.

To construct our ID-based authenticated key exchange (AKE) scheme, we combine this CCA-secure ID-KEM with a CPA-secure KEM (in our case, the CPA-secure version of NewHope-512 [4]), similar to the generic ID-AKE construction in [25]. We profiled our ID-AKE protocol, shown in Fig. 3(a), on the same platform mentioned in Section III-D, and our hardware-accelerated implementation takes 9.25 ms and consumes 57.43 μJ energy at 1.1 V and 72 MHz. The corresponding ID-based TLS handshake is shown in Fig. 3(b). Since the client and the server are respectively the initiator and the responder in our protocol, the key shares in *ClientHello* and *ServerHello* are (pk, c_1) and (c, c_2) respectively, and the corresponding shared secret is ss. Since the ID-KEM is used for authentication, the *CertificateRequest*, *Certificate* and *CertificateVerify* messages can be omitted altogether. The total communication cost of our proposed ID-based certificate-less post-quantum TLS handshake is 9,731 bytes, which is $4.5\times$ smaller than certificate-based post-quantum TLS handshake at similar security level.

We compare the total client-side energy consumption (computation and communication) for pre-quantum and post-quantum TLS 1.3 handshakes, both traditional certificate-based and certificate-less ID-based. Since public key cryptog-

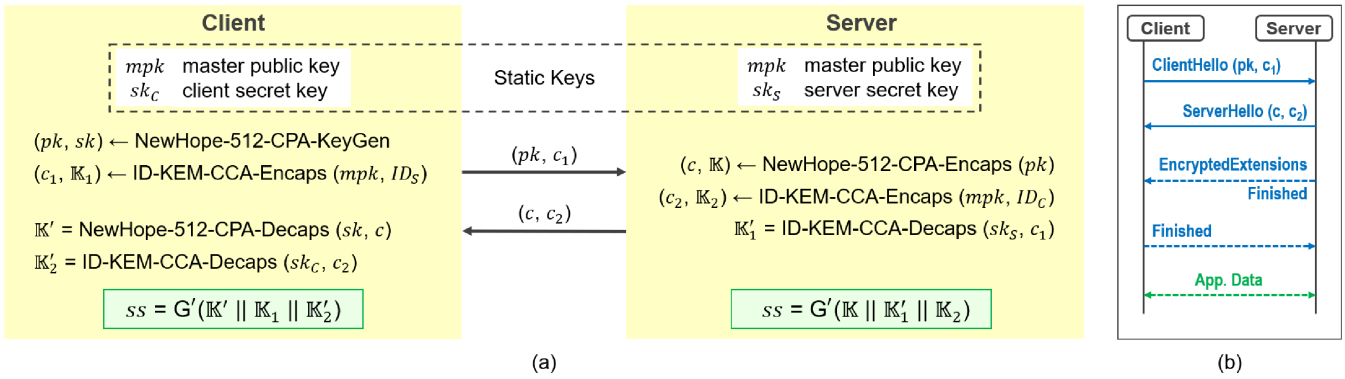

Fig. 3. Our proposed (a) ID-based authenticated key exchange scheme from Ring-LWE and (b) corresponding ID-based TLS handshake.

TABLE IV
TLS 1.3 HANDSHAKE COMPUTATION AND COMMUNICATION COSTS ON THE CLIENT SIDE (CERTIFICATE-BASED AND ID-BASED)

	Pre-Quantum (pairing-based)		Post-Quantum (lattice-based)	
	cert	ID	cert	ID
Handshake (bytes)	1,820	547	43,452	9,731
Comp. Time (ms) †	175.27	2992.88	14.69	27.11
Comp. Energy (μJ) †	148.87	2621.43	36.60	57.43
Comm. Time (ms)	14.56	4.38	347.62	77.85
Comm. Energy (μJ)	41.5	12.53	990.39	221.61

† All computation time and energy normalized at 20 MHz and 1.1 V

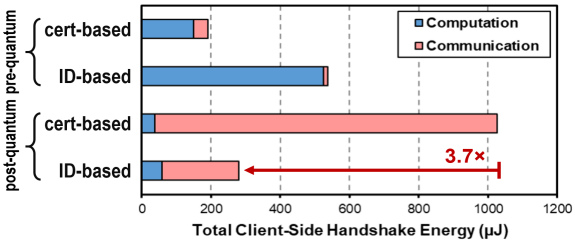

Fig. 4. Total client-side energy (with hardware-accelerated cryptographic computation; communication over Bluetooth Low Energy) of pre- and post-quantum TLS 1.3 handshake, both certificate-based and ID-based.

raphy accounts for 99% of TLS handshake computations [26], we consider the total handshake compute energy to be equal to that of the AKE protocol. To better understand the impact of communication cost reduction in ID-based TLS, we consider only hardware-accelerated cryptography computations since most embedded micro-controllers have dedicated hardware for standard cryptographic primitives. For certificate-based pre-quantum TLS with ECDHE and ECDSA, the compute energy is obtained from [26]. For ID-based pre-quantum TLS with ECDHE and elliptic curve pairing-based ID-KEM [24], the compute costs are from [26] and [27]. For certificate-based post-quantum TLS with NewHope-512-CPA-KEM and qTesla-I, we refer to [14] for the computation costs. Finally, these are compared with our ID-based post-quantum TLS handshake implemented on the same platform as [14]. For all communications, we consider a 1 Mbps Bluetooth Low Energy link and refer to the state-of-the-art transceiver in [28] for power numbers. All these results are summarized in Table IV, and Fig. 4 shows the total client-side handshake energy consumption. Clearly, pre-quantum TLS is dominated by computation costs even after cryptographic hardware acceleration, while handshake communications dominate post-quantum TLS with hardware-accelerated cryptography. In fact, ID-based TLS is 2.8× costlier than certificate-based TLS in the pre-quantum scenario since pairing computations are an order of magnitude more expensive than traditional ECC [27]. However, in the post-quantum case, ID-based TLS provides a clear advantage over using certificates, with 3.7× reduction in total handshake energy consumption.

V. CONCLUSION AND FUTURE WORK

In this work, we demonstrate quantum-secure ID-based CCA-secure encryption, key encapsulation and authenticated key exchange from lattices, based on the CPA-secure DLP-IBE scheme. We propose secure and efficient parameters and also provide implementation results. We integrate this key exchange with the TLS 1.3 protocol to allow certificate-less authentication. Comparison of total post-quantum TLS handshake costs (with hardware-accelerated cryptography) shows that our proposed ID-based scheme is 3.7× more energy-efficient than traditional certificate-based authentication. Our CCA-secure IBE scheme can also be used to implement different post-quantum network security protocols for WSNs.

ACKNOWLEDGMENT

The authors thank Texas Instruments for funding this work.

REFERENCES

[1] L. B. Oliveira et al., "Identity-Based Encryption for Sensor Networks," in *IEEE Int. Conference on Pervasive Computing and Commun. Workshops (PerComW)*, pp. 290-294, Mar. 2007.
[2] G. Alagic et al., "Status Report on the First Round of the NIST Post-Quantum Cryptography Standardization Process," *NIST Technical Report*, no. 8240, Jan. 2019.
[3] C. Peikert, "A Decade of Lattice Cryptography," in *Now Publishers – Foundations and Trends in Theoretical Computer Science*, vol. 10, no. 4, pp. 283-424, Mar. 2016.
[4] T. Poppelmann et al., "NewHope – Algorithm Specifications and Supporting Documentation," *NIST Technical Report*, 2019.
[5] N. Bindel et al., "qTESLA – Algorithm Specifications and Supporting Documentation," *NIST Technical Report*, 2019.
[6] NIST, "Advanced Encryption Standard (AES)," *NIST Technical Report*, FIPS PUB 197, Nov. 2001.
[7] NIST, "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions," *NIST Technical Report*, FIPS PUB 202, Aug. 2015.
[8] C. Gentry, C. Peikert and V. Vaikuntanathan, "Trapdoors for Hard Lattices and New Cryptographic Constructions," in *ACM Symp. on Theory of Computing (STOC)*, pp. 197-206, May 2008.
[9] L. Ducas et al., "Efficient Identity-Based Encryption over NTRU Lattices," in *IACR ASIACRYPT*, pp. 22-41, Dec. 2014.
[10] S. McCarthy et al., "A Practical Implementation of Identity-Based Encryption Over NTRU Lattices," in *IMA Int. Conf. on Cryptography and Coding (IMACC)*, pp. 227-246, Dec. 2017.
[11] T. Guneysu and T. Oder, "Towards Lightweight Identity-Based Encryption for the Post-Quantum-Secure Internet of Things," in *Int. Symp. on Quality Electronic Design (ISQED)*, pp. 319-324, Mar. 2017.
[12] D. Hofheinz et al., "A Modular Analysis of the Fujisaki-Okamoto Transformation," in *Theory of Crypto. (TCC)*, pp. 341-371, Nov. 2017.
[13] U. Banerjee et al., "An Energy-Efficient Configurable Lattice Cryptography Processor for the Quantum-Secure Internet of Things," in *IEEE Int. Solid-State Circuits Conf. (ISSCC)*, pp. 46-48, Feb. 2019.
[14] U. Banerjee, T. S. Ukyab and A. P. Chandrakasan, "Sapphire: A Configurable Crypto-Processor for Post-Quantum Lattice-based Protocols," in *IACR Trans. on Cryptographic Hardware and Embedded Systems (TCHES)*, vol. 2019, no. 4, pp. 17-61, Aug. 2019.
[15] M. R. Albrecht et al., "On the Concrete Hardness of Learning with Errors," in *J. of Math. Crypto.*, vol. 9, no. 3, pp. 169-203, Oct. 2015.
[16] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," *IETF RFC 8446*, Aug. 2018.
[17] U. Banerjee et al., "eeDTLS: Energy-Efficient Datagram Transport Layer Security for the Internet of Things," in *IEEE Global Commun. Conf. (GLOBECOM)*, pp. 1-6, Dec. 2017.
[18] J. W. Bos et al., "Post-Quantum Key Exchange for the TLS Protocol from the Ring Learning with Errors Problem," in *IEEE Symp. on Security and Privacy*, pp. 553-570, May 2015.
[19] X. Gao et al., "Efficient Implementation of Password-Based Authenticated Key Exchange from RLWE and Post-Quantum TLS," in *IACR Cryptology ePrint Archive*, Report 2017/1192, Dec. 2017.

[20] E. Crockett et al., "Prototyping Post-Quantum and Hybrid Key Exchange and Authentication in TLS and SSH," in *NIST 2nd PQC Standardization Conference*, Aug. 2019.
[21] J. Sepulveda et al., "Post-Quantum Enabled Cyber Physical Systems," in *IEEE Embedded Sys. Letters*, vol. 11, no. 4, pp. 106-110, Dec. 2019.
[22] P. Kampanakis et al. "The Viability of Post-Quantum X.509 Certificates," in *IACR Cryptology ePrint Archive*, Report 2018/063, Jan. 2018.
[23] C. Peng et al., "Improved TLS Handshake Protocols using Identity-Based Cryptography," in *Int. Symp. on Information Engineering and Electronic Commerce*, pp. 135-139, May 2009.
[24] K. Bentahar et al., "Generic Constructions of Identity-Based and Certificateless KEMs," in *Journal of Cryptology*, vol. 21, no. 2, pp. 178-199, Apr. 2008.
[25] A. Fujioka et al., "Strongly Secure Authenticated Key Exchange from Factoring, Codes, and Lattices," in *Int. Workshop on Public Key Cryptography (PKC)*, pp. 467-484, May 2012.
[26] U. Banerjee et al., "An Energy-Efficient Reconfigurable DTLS Cryptographic Engine for End-to-End Security in IoT Applications," in *IEEE Int. Solid-State Circuits Conf. (ISSCC)*, pp. 42-44, Feb. 2018.
[27] T. Unterluggauer and E. Wenger, "Efficient Pairings and ECC for Embedded Systems," in *Int. Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, pp. 298-315, Sep. 2014.
[28] H. Liu et al., "An ADPLL-centric Bluetooth Low-Energy Transceiver with 2.3mW Interference-Tolerant Hybrid-Loop Receiver and 2.9mW Single-Point Polar Transmitter in 65nm CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC)*, pp. 444-446, Feb. 2018.

APPENDIX A
SAGE CODE FOR PARAMETER SELECTION

Here, we provide the Sage code we have used to determine various parameters for our lattice-based IBE implementation.

A. Security Analysis with Different Error Distributions

To estimate security for $(n, q) = (1024, 8380417)$ and uniform error distribution over $\{-1, 0, 1\}$:

```
load("https://bitbucket.org/malb/
      lwe-estimator/raw/HEAD/
      estimator.py")

n = 1024; q = 8380417
stddev = sqrt(2/3)
alpha = sqrt(2.0*pi)*stddev/q
_ = estimate_lwe(n, alpha, q,
    secret_distribution=(-1,1),
    reduction_cost_model=BKZ.sieve)
```

To estimate security for $(n, q) = (1024, 8380417)$ and trinary error distribution over $\{-1, 0, 1\}$ with probability parameter $\rho \in \{1/2, 1/4, 1/8\}$:

```
load("https://bitbucket.org/malb/
      lwe-estimator/raw/HEAD/
      estimator.py")

n = 1024; q = 8380417
for i in range(3):
    rho = 1/2**(i+1)
    stddev = sqrt(rho)
    alpha = sqrt(2.0*pi)*stddev/q
    print("i = %d" % (i+1))
    _ = estimate_lwe(n, alpha, q,
        reduction_cost_model=BKZ.sieve)
```

For both cases, we have used the commit version `e46ac6607a25b2aaada76eaa1515f0b6a7a35889` of the online Sage-based LWE hardness estimator tool https://bitbucket.org/malb/lwe-estimator which was accessed on 21st September 2019 for our calculations.

B. Number Theoretic Transform (NTT) Parameters

To determine ω and ψ, respectively the smallest n-th and $2n$-th roots of unity modulo q:

```
n = 1024; q = 8380417
R = IntegerModRing(q); g = R(1)

r = g.nth_root(n, all=True)
r.sort()
omega = 1
for root in r:
    count = 0
    for i in range(1,n):
        if root**i % q == 1:
            count = count + 1
    if count == 0:
        omega = root
        break
print("omega = %d" % omega)

r = g.nth_root(2*n, all=True)
r.sort()
psi = 1
for root in r:
    count = 0
    for i in range(1,2*n):
        if root**i % q == 1:
            count = count + 1
    if count == 0:
        psi = root
        break
print("psi   = %d" % psi)
```

A revised version of this paper was published in 2020 IEEE International Conference on Communications (ICC) - DOI: 10.1109/ICC40277.2020.9148829