Performance Evaluation of Secure Routing Protocols with Lattice Signature Scheme for Wireless Multi-hop Networks

Ukyo Hasegawa* Hideharu Kojima**
Faculty of Information Science and Technology
Osaka Institute of Technology
Hirakata, Japan
*e1n21077@oit.ac.jp **hideharu.kojima@oit.ac.jp

***Abstract*—For a few decades, routing protocols for wireless multi-hop networks have been investigated extensively. One of the most important roles of routing protocols is route establishment between nodes. To establish trusted routes, secure routing protocols also have been extensively researched. In this paper, we focus on these secure routing protocols that equip authentication mechanisms, such as digital signatures, to protect against attacks injecting fraudulent data into the route information in the process of route establishment. We apply an ID-based aggregate signature scheme based on lattice to the signature algorithm of the proposed secure routing protocol. We call its resulting protocol *ISDSR-L*. To evaluate our proposed protocol, we have conducted experiments measuring route establishment times on emulation environments, signature generation time, and signature verification time. Regarding computational efficiency, while our protocol demonstrates sufficient performance, route establishment times are slightly slower than a comparison that adopts the ECDSA scheme in its signature scheme.**

***Index Terms*—secure routing protocols, ID-based aggregate signatures, signatures based on lattice, performance evaluation**

I. Introduction

For a few decades, routing protocols for wireless multi-hop networks have been investigated extensively. One feature of wireless multi-hop networks is that anyone can dynamically join/leave into/from the network. In addition, anyone can receive/send data from/to the network within the propagation range of each node. This means that an attacker can send malicious and fraudulent data to the network, such as fake route information [1]. In [2], malicious devices can make packet loops by injecting fake route information. To protect against such attacks, routing protocols for trusted route establishment are needed. To achieve trusted route establishment, secure routing protocols have been investigated [3]–[5]. The existing secure routing protocols adopt authentication mechanisms, such as digital signatures, to prevent the attackers from injecting fraudulent data into the route information in the process of route establishment [3], [5]–[11]. These protocols guarantee the validity of the route information. In these protocols, after forwarding nodes receive a packet, they verify the signature in the received packet whether the signature is valid or not. If it is valid, they add their ID to the route information and generate signatures from the route information. Afterward, they send the route information and the generated signatures as a packet. Every forwarding node attaches their generated signatures to their forwarding packet. The total signature length in the packet increases linearly. To suppress the total signature length, several works [5], [7]–[9] adopt a multi-signature scheme [12], [13], where individual signatures generated by a group of signers can be combined into a single short signature. These works have a promising feature: they can utilize ID information for each device as a public key and maintain a constant signature size regardless of the number of devices. Using RSA or ECDSA is also a common choice for implementing digital signature schemes, including aggregate signatures. However, in recent years, a digital signature scheme based on lattice [14] has also emerged as a viable alternative. In anticipation of the quantum computing era, in order to defend against attacks, we believe that secure routing protocols should incorporate signature schemes based on lattice. Besides, a signature scheme based on the lattice is equal to or better than ECDSA in terms of the computational time for signature generation and verification [15]. It is expected that it can improve security as well as accelerating signature generation and verification processes.

Therefore, we design an ID-based aggregate signature scheme based on the lattice and its resultant secure routing protocol, called *ISDSR-L*. In our previous work [16], we evaluated calculation time for the signature generation and verification of our previous proposed ID-based aggregate signature algorithm with lattice on a Raspberry Pi 4 and a laptop PC. In this paper, we implemented our secure routing protocol named *ISDSR-L*, which adopted an ID-based aggregate signature algorithm with lattice. In addition, we evaluate the route establishment time of *ISDSR-L* comparing with a secure routing protocol with ECDSA in emulation environments. We believe that our evaluation results contribute to a better understanding of the performance of protocols utilizing state-of-the-art cryptographic theory.

979-8-3315-2096-0/25/$31.00 ©2025 IEEE

II. Related Works

Many researches have been conducted in the area of wireless multi-hop networks. In this section, we describe research for secure routing protocols that guarantee the validity of route information between a source node and its destination node in wireless multi-hop networks.

A. Secure Routing Protocols

The first secure routing protocol [1] was a key management protocol for ad hoc networks. Next, Hu et al. [3], [17] proposed Ariadne with message authentication codes (MAC) and digital signatures. Since digital signatures are able to provide the non-repudiation of signers, we focus on digital signatures. Since standard digital signatures need certificates of public keys to bind public keys and their owners, the existing protocols [6]–[8], [18]–[20] also need a public key infrastructure. As a more advanced approach, there are several protocols [21], [22] utilizing ID-based cryptography [23], which allows users to utilize any string as a public key. This approach is more efficient because they do not need a public key infrastructure. The above existing protocols utilize standard signature schemes for single-user settings where the bit length of the signatures increases linearly in the number of signers. ISDSR+ [5] is the first protocol that uses ID-based cryptography and fixed-length signatures. Our proposed scheme in this paper adopts a signature scheme based on lattice as post-quantum cryptography.

B. Aggregate Signature Based on Lattice

In recent years, aggregate signature schemes based on lattice have been investigated [24]–[27]. In the paper [27], an aggregate signature scheme based on Crystal-Dilithium [14], which is based on lattice, was proposed. Our proposed scheme differs from these schemes with respect to an ID-based scheme.

Our scheme can utilize ID as a public key. ID-based aggregate signature schemes based on lattice [28], [29] also have been investigated. The scheme in the paper [29] designates the number of signatures included in an aggregate signature in advance. It means a restriction on the number of nodes within a route, and hence, it is difficult to apply this scheme to wireless multi-hop networks. The scheme in the paper [28] communicates with a key manager several times when a node generates and verifies a signature. In wireless multi-hop networks, it is considered that the number of communications with a key manager should be less. In our previous scheme [16], a node communicates with a key manager only once during the key derivation process. However, the size of the aggregate signature is not constant. Every node adds a matrix for signature verification. In our current scheme of this paper, we improve our previous scheme to make the size of the aggregate signature constant.

III. ISDSR-L (ISDSR with Lattice)

The secure DSR with ID-based sequential aggregate signatures with lattice (ISDSR-L) is a secure routing protocol with a multi-signature scheme to guarantee the validity of route information between a source node and its destination node. More specifically, ISDSR-L is based on an ID-based sequential aggregate signature scheme [30], where each user generates a single signature by taking both messages to be signed and a signature-so-far as input as well as utilizing any string as its own public key. In this section, we first recall the ID-based sequential aggregate signature scheme based on lattice and then briefly describe a route establishment process of ISDSR-L.

Algorithm 1 Setup

function SETUP
  $\rho \leftarrow \{0,1\}^{256}$, $K \leftarrow \{0,1\}^{256}$
  $(\mathbf{s}_1, \mathbf{s}_2) \leftarrow S_\eta^k \times S_\eta^k$
  $\mathbf{A} \in R_q^{k \times k}$:= $\texttt{ExpandA}(\rho)$
  $\mathbf{t} := \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2$
  $tr \in \{0,1\}^{384}$:= $\texttt{CRH}(\rho || \mathbf{t})$
  return $(mpk = (\rho, \mathbf{t}), msk = (\rho, K, tr, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t})$
end function

Algorithm 2 KeyDerivation

Require: $msk = (\rho, K, tr, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t})$, ID
function KEYDERIVATION
  $\mathbf{t}_{id} := \mathbf{t} \cdot Hash(ID)$
  $\mathbf{sk}_{id2} := \mathbf{s}_2 \cdot \texttt{Uniform}(Hash(ID))_\eta$
  $\mathbf{A} \in R_q^{k \times k}$:= $\texttt{ExpandA}(\rho)$
  $inv_{det} := \texttt{invmod}_q(|\mathbf{A}|)$
  $\tilde{\mathbf{A}} := \texttt{adjugate}_q(\mathbf{A})$
  $\mathbf{sk}_{id1} := inv_{det}\tilde{\mathbf{A}}(\mathbf{t}_{id} - \mathbf{sk}_{id2})$
  $tr_{id} \in \{0,1\}^{384}$:= $\texttt{CRH}(\rho || \mathbf{t}_{id})$
  return $sk_{id} = (\rho, K, tr_{id}, \mathbf{sk}_{id1}, \mathbf{sk}_{id2})$
end function

A. Building Blocks: ID-based Aggregate Signatures Based on Lattice

The ID-based sequential aggregate signature scheme based on lattice for ISDSR-L consists of **Setup**, **KeyDerivation**, **Sign**, and **Verification**, which are shown in **Algorithms 1**-**4**, respectively. This scheme is an extension of the signature algorithm of Crystals-Dilithium [14]. In these algorithms, a left arrow ($\leftarrow$) denotes a random choice of an element from the given group, and an equal symbol (:=) denotes a substitution of the result of a deterministic computation on the right-hand side to the left-hand side. We follow Crystals-Dilithium. The regular font letters represent elements in R or R_q, including elements in $\mathbb{Z}$ and $\mathbb{Z}_q$. The bold lower-case letters denote column vectors with coefficients in R or R_q. By default, all vectors will be column vectors. The bold upper-case letters represent matrices. We also use uniform sampling shown in [31]–[33] in the same manner as Crystals-Dilithium. Functions in these algorithms, i.e., `ExpandA`, `CRH`, `ExpandMask`, `H`, `Decompose`, and `HighBits`, are identical to those in Crystals-Dilithium.

Algorithm 1 generates a master public key msk and a master secret key msk. Algorithm 2 generates an ID-base

Algorithm 3 Signing

Require: sk_{id}, M, σ'

1: **function** SIGNING
2: Parse sk_{id}=(ρ, K, tr_{id}, $\mathbf{sk}_{id1}$, $\mathbf{sk}_{id2}$)
3: $\mathbf{A} \in R_q^{k \times k}$:= ExpandA(ρ)
4: $\mu \in \{0,1\}^{384}$:= CRH ($tr_{id}||M$)
5: κ := 0, $\mathbf{z}$:= $\perp$
6: **while** $\mathbf{z} = \perp$ **do**
7: $\mathbf{y} \in S^k_{\gamma_1 - 1}$:= ExpandMask($K||\mu||\kappa$)
8: $\mathbf{w}$:= $\mathbf{Ay}$
9: $\mathbf{w}^1$:= $\mathtt{HighBits}_q(\mathbf{w}, 2\gamma_2)$
10: $c \in B_{60}$:= H (μ)
11: $\mathbf{z}$:= $\mathbf{y}$ + $c\mathbf{sk}_{id1}$
12: ($\mathbf{r}^1$, $\mathbf{r}^0$) := $\mathtt{Decompose}_q(\mathbf{w} - c\mathbf{sk}_{id2}, 2\gamma_2)$
13: **if** $||\mathbf{r}^0||_\infty \geq \gamma_2 - \beta$ or $\mathbf{r}^1 \neq \mathbf{w}^1$ **then** $\mathbf{z} := \perp$
14: **end if**
15: $\kappa := \kappa + 1$
16: **end while**
17: Parse $\sigma' = (c', \mathbf{z}', \mathbf{w}')$
18: **if** $id = 1$ **then**
19: $\sigma' := (c' = 0, \mathbf{z}' = \mathbf{0}, \mathbf{w}' = \mathbf{0})$
20: **end if**
21: $c_{agg} := c' + c$, $\mathbf{z}_{agg} := \mathbf{z}' + \mathbf{z}$, $\mathbf{w}_{agg} := \mathbf{w}' + \mathbf{w}$
22: **return** $\sigma = (c_{agg}, \mathbf{z}_{agg}, \mathbf{w}_{agg})$
23: **end function**

Algorithm 4 Verification

Require: σ, mpk, List $((id_1, m_1), ..., (id_n, m_n))$ of ID Info. and Message

1: **function** VERIFICATION
2: Parse mpk = (ρ, $\mathbf{t}$)
3: Parse σ := (c_{agg}, $\mathbf{z}_{agg}$, $\mathbf{w}_{agg}$)
4: c_N := 0, $\mathbf{ct}_N$
5: **for** i := 1; $i \leq n$; i++ **do**
6: $\mathbf{t}_{id}$:= $\mathbf{t} \cdot Hash(id_i)$
7: tr := CRH($\rho||\mathbf{t}_{id}$)
8: μ := CRH($tr||m_i$)
9: c := H(μ)
10: c_N := c_N + c
11: $\mathbf{ct}_N$:= $\mathbf{ct}_N$ + $c\mathbf{t}_{id}$
12: **end for**
13: $\mathbf{A} \in R_q^{k \times k}$:= ExpandA(ρ)
14: $\mathbf{Az}^1$:= $\mathtt{HighBits}_q(\mathbf{Az}_{agg} - \mathbf{ct}_N, 2\gamma_2)$
15: $\mathbf{w}^1_{agg}$:= $\mathtt{HighBits}_q(\mathbf{w}_{agg}, 2\gamma_2)$
16: **return** $c_N = c_{agg}$ && $\mathbf{w}^1_{agg} = \mathbf{Az}^1$
17: **end function**

secret key sk_{id}. All nodes execute this function to generate their own secret key. The generated secret key consists of five elements, i.e., ρ, K, tr_{id}, $\mathbf{sk}_{id1}$, and $\mathbf{sk}_{id2}$. The relationship between these keys satisfy the equation below.

$$\mathbf{t} \cdot Hash(ID) \equiv \mathbf{t}_{id} \equiv \mathbf{Ask}_{id1} + \mathbf{sk}_{id2}. \tag{1}$$

$\mathbf{sk}_{id2}$ is computed on Line 3 of Algorithm 2. To compute $\mathbf{sk}_{id1}$, we transform Equation (1) to Equation (2) and Equation (3) as a matrix description.

$$\mathbf{t}_{id} - \mathbf{sk}_{id2} \equiv \mathbf{Ask}_{id1}. \tag{2}$$

$$\begin{bmatrix} t_{id,1} - sk_{id2,1} \\ t_{id,2} - sk_{id2,2} \\ \cdots \\ t_{id,k} - sk_{id2,k} \end{bmatrix} \equiv \begin{bmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,k} \\ A_{2,1} & A_{2,2} & \cdots & A_{2,k} \\ \cdots & & & \\ A_{k,1} & A_{k,2} & \cdots & A_{k,k} \end{bmatrix} \begin{bmatrix} sk_{id1,1} \\ sk_{id1,2} \\ \cdots \\ sk_{id1,k} \end{bmatrix}. \tag{3}$$

The left-hand side $\mathbf{t}_{id} - \mathbf{sk}_{id2}$ and the right-hand side $\mathbf{Ask}_{id1}$ are congruent under the modulo q. $\mathbf{sk}_{id1}$ is derived to solve the congruence equation (2). To solve this, an inverse element (inv_{det}) of the determinant $\mathbf{A}$ is computed at Line 5 of Algorithm 2. At Line 6, an adjugate matrix ($\tilde{\mathbf{A}}$) of $\mathbf{A}$ is computed. Finally, $\mathbf{sk}_{id1}$ is derived from Equation (4) and Equation (5).

$$inv_{det}\tilde{\mathbf{A}}(\mathbf{t}_{id} - \mathbf{sk}_{id2}) \equiv inv_{det}\tilde{\mathbf{A}}\mathbf{Ask}_{id1} \tag{4}$$

$$inv_{det}\tilde{\mathbf{A}}(\mathbf{t}_{id} - \mathbf{sk}_{id2}) \equiv \mathbf{sk}_{id1} \tag{5}$$

Next, Algorithm 3 and Algorithm 4 are described below. There are three nodes, i.e., v_a, v_b, and v_c. Each node has mpk and msk, and generates its own secret key using Algorithm 2. v_a generates a signature σ_a on route information (a) as a message as follows:

$$\sigma_a = (c_{agg} = c_a, \mathbf{z}_{agg} = \mathbf{z}_a, \mathbf{z}_{agg} = \mathbf{w}_a). \tag{6}$$

Then, v_a sends the route information (a) and the signature σ_a to the next node v_b. Then, v_b adds its own information into the route information after v_b receives the route information (a) and the signature σ_a. v_b also generates a signature σ_b from route information (a,b) as a message as follows:

$$\sigma_b = (c_{agg} = c_a + c_b, \mathbf{z}_{agg} = \mathbf{z}_a + \mathbf{z}_b, \mathbf{w}_{agg} = \mathbf{w}_a + \mathbf{w}_b). \tag{7}$$

v_c then receives the route information (a,b) and the signature σ_b. v_c verifies this signature using Algorithm 4. After each c_N and $\mathbf{ct}_N$ is calculated. The verification process reaches Line 16 in Algorithm 4. The left-hand side of Equation (8) is equal to the right-hand side because $\mathbf{z}_{agg}$ represents $\mathbf{z}_a$ + $\mathbf{z}_b$ and $\mathbf{ct}_N$ also represents $c_a\mathbf{t}_a$ + $c_b\mathbf{t}_b$. Each value is represented as follows:

$$\mathbf{Az}_{agg} - \mathbf{ct}_N = \mathbf{A}(\mathbf{z}_a + \mathbf{z}_b) - (c_a\mathbf{t}_a + c_b\mathbf{t}_b), \tag{8}$$

$$\begin{aligned} \mathbf{Az}_a &= \mathbf{Ay}_a + c_a\mathbf{Ask}_{a1} \\ \mathbf{Az}_b &= \mathbf{Ay}_b + c_b\mathbf{Ask}_{b1}, \end{aligned} \tag{9}$$

$$\begin{aligned} c_a\mathbf{t}_a &= c_a\mathbf{Ask}_{a1} + c_a\mathbf{sk}_{a2} \\ c_a\mathbf{t}_b &= c_b\mathbf{Ask}_{b1} + c_b\mathbf{sk}_{b2}, \end{aligned} \tag{10}$$

$$\mathbf{Az}_{agg} - \mathbf{ct}_N = \mathbf{Ay}_a + \mathbf{Ay}_b - c_a\mathbf{sk}_{a2} - c_a\mathbf{sk}_{b2}, \tag{11}$$

$$\mathbf{w}_a = \mathbf{Ay}_a, \mathbf{w}_b = \mathbf{Ay}_b, \tag{12}$$

$$\mathbf{Az}_{agg} - \mathbf{ct}_N = \mathbf{w}_a - c_a\mathbf{sk}_{a2} + \mathbf{w}_b - c_a\mathbf{sk}_{b2}. \tag{13}$$

Equation (9) is derived from Line 11 in Algorithm 3 and Equation (10) is derived from Equation (1). Applying Equation (9) and Equation (10), Equation (8) becomes Equation (11).

Algorithm 5 Decompose_q

Require: r, α
1: **function** DECOMPOSE
2: $r := r \bmod q$
3: $r^0 := r \bmod \alpha$
4: **if** r - r^0 = q -1 **then**
5: $r^1 := 0$, $r^0 := r^0 - 1$
6: **else**
7: $r^1 := (r - r^0)/\alpha$
8: **end if**
9: **return** (r^1, r^0)
10: **end function**

Algorithm 6 HighBits_q

Require: r, α
1: **function** HIGHBITS
2: $(r^1, r^0) := \texttt{Decompose}_q(r, \alpha)$
3: **return** r^1
4: **end function**

Finally, Equation (11) is transformed to Equation (13) to substitute $\mathbf{w}_a$ and $\mathbf{w}_b$ for $\mathbf{A}\mathbf{y}_a$ and $\mathbf{A}\mathbf{y}_b$, respectively. From Line 9 and Line 12 in Algorithm 3, $\mathbf{w}_a$ and $\mathbf{w}_a - c\mathbf{sk}_{a2}$ are represented as follows:

$$\begin{aligned} \mathbf{w}_a^1 &:= \texttt{HighBits}_q(\mathbf{w}_a, 2\gamma_2), \\ (\mathbf{r}_a^1, \mathbf{r}_a^0) &:= \texttt{Decompose}_q(\mathbf{w}_a - c\mathbf{sk}_{id2}, 2\gamma_2). \end{aligned} \tag{14}$$

As described in Algorithm 5 and Algorithm 6, the return value r^1 from $\texttt{Decompose}_q$ and the return value from $\texttt{HighBits}_q$ are equal. Because of that, $\mathbf{w}_a^1$ and $\mathbf{r}_a^1$ are equal to each other. Line 13 in Algorithm 3 indicates if $\mathbf{r}^1$ and $\mathbf{w}^1$ are unequal then $\mathbf{z}$ becomes $\perp$ and the signing process goes to Line 7 and generates $\mathbf{z}$ and c again. As a result, Equation (15) becomes true from Line 14 to 16 in Algorithm 4.

$$\texttt{HighBits}_q(\mathbf{w}_{agg}, 2\gamma_2) = \texttt{HighBits}_q(\mathbf{A}\mathbf{z}_{agg} - \mathbf{ct}_N, 2\gamma_2). \tag{15}$$

B. Protocol Overview

ISDSR-L uses three kinds of packets, i.e., $SRREQ$, $SRREP$, and $SRERR$. $SRREQ$ and $SRREP$ are used in the *secure route discovery* phase to establish a connection from a source node to its destination node. $SRERR$ is used in the *secure route maintenance* phase in which a node finds a disconnection of the destination. In both phases, packets include signatures for route information between a source and its destination. We describe in detail the *secure route discovery* phase below. There are three nodes, a source node v_s, its destination node v_d, and a forwarding node v_f. Each node utilizes its own ID, e.g., IP address or device name, as a public key. $SRREQ$ and $SRREP$ are represented as $SRREQ = (src, dest, ri, \sigma)$ and $SRREP = (src, dest, ri, \sigma)$, src is a source node, $dest$ is src's destination node, ri is route information, and σ is an aggregate signature.

Step 1: v_s generates $SRREQ_s$ to establish a route to v_d.
v_s makes route information $ri(s)$ that contains the ID of v_s and also generates a signature $\sigma_s = (c_s, \mathbf{z}_s, \mathbf{w}_s)$ by **Signing** described in Algorithm 3. Then, v_s broadcasts $SRREQ_s = (v_s, v_d, ri(s), \sigma_s)$.

Step 2: v_f forwards a received $SRREQ$.
v_f receives $SRREQ_s$ from v_s and verifies the signature σ_s through **Verification** (Algorithm 4). If it is true, v_f checks whether the destination of the packet is itself or not. Since v_f is not the destination of the packet, v_f adds own ID to ri, then $ri(s)$ becomes $ri(s, f)$. v_f generates $\sigma_f = (c_f, \mathbf{z}_f, \mathbf{w}_f)$ and aggregates σ_s and σ_f as $\sigma_{sf} = (c_s + c_f, \mathbf{z}_s + \mathbf{z}_f, \mathbf{w}_s + \mathbf{w}_f)$ by **Signing** described in Algorithm 3. Then, v_f broadcasts $SRREQ_f = (v_s, v_d, ri(s, f), \sigma_{sf})$.

Step 3: v_d replies $SRREP_d$ to v_s.
v_d receives $SRREQ_f$ from v_f and verifies the signature σ_{sf} through **Verification** (Algorithm 4). If it is true, v_d checks whether the destination of the packet is itself or not. Since v_d is the destination of the packet, v_d generates $SRREP$. v_d adds own ID to ri, then $ri(s, f)$ becomes $ri(s, f, d)$. v_d generates $\sigma_d = (c_d, \mathbf{z}_d, \mathbf{w}_d)$ and aggregates σ_{sf} and σ_d as $\sigma_{sfd} = (c_s + c_f + c_d, \mathbf{z}_s + \mathbf{z}_f + \mathbf{z}_d, \mathbf{w}_s + \mathbf{w}_f + \mathbf{w}_d)$ by **Signing** described in Algorithm 3. v_d sends $SRREP_d = (v_d, v_s, ri(s, f, d), \sigma_{sfd})$ to v_f.

Step 4: v_s receives $SRREP_d$ from v_d.
v_f verifies σ_{sfd} in the received $SRREP_d$ and forwards it to v_s. v_s also verifies σ_{sfd} in the received $SRREP_d$ from v_f. If it is true, a route between v_s and v_d is established.

IV. EXPERIMENTS

In this section, we conduct experiments and show their results.

A. Setting

We implemented our proposed secure routing protocol that equips an ID-based aggregate signature scheme based on the lattice in C++ language. We have released our implementation via GitHub[1]. This implementation is based on the Crystals-dilithium library[2]. As the first step of the performance evaluation for ISDSR-L, we also utilize the implementation of ISDSR+ [5] in the C++ language and this protocol adopts an aggregate signature scheme with ECDSA. The implementation of ISDSR+ utilizes the mcl library[3] as the elliptic curve library.

We use UTM as a virtual machine for running Ubuntu 20.04 on Mac Book Pro. The experiments are conducted in the emulation environments on the Mininet-Wifi [34], a network emulator on Ubuntu 20.04. Mininet-Wifi provides a virtual wireless environment where the user can set the network topology, the communication range of the terminal, etc. It also allows the user to run any programs on the virtual

[1]Available at https://github.com/kjm-hdhr/isdsr_cpp.
[2]Available at https://github.com/pq-crystals/dilithium.
[3]Available at https://github.com/herumi/mcl.

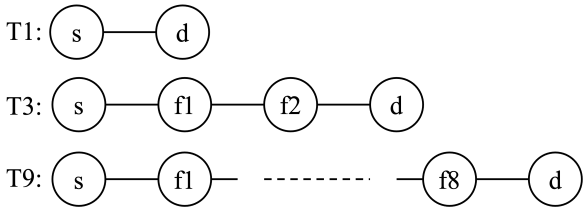

Fig. 1. Illustration of Network Topologies in Experiments

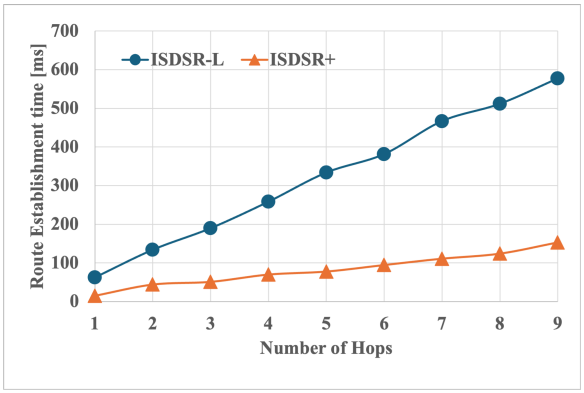

Fig. 2. Route Establishment Time [ms]

terminal. The radio propagation model used in the emulation environments is logDistance with exp = 4.5. The wireless LAN standard was 802.11g. The experiments described below are conducted in the following environment:

- Mac Book Pro M2 Max (Ventura 13.2, 96GB MEM).
- UTM ver. 4.5.3 (virtual machine)
- Ubuntu 20.04
- Mininet-wifi ver. 2.6

We prepare nine line shape network topologies representing one-hop to nine-hop. For examples, the topology T1 in Fig. 1 represents one-hop topology in which a source node communicates with its destination directly. The topology T3 and T9 in Fig. 1 represent three-hop topology and nine-hop topology, respectively. T3 consists of one source node, one destination node, and two forwarding nodes. T9 consists of one source node, one destination node, and eight forwarding nodes. In each network, a source node starts a route establishment process toward its destination, and all nodes make packet operations to establish a route, as mentioned in section III-B. We conducted the measurements of the route establishment time for each of the nine network topologies, and each result represents the average time over 100 runs.

B. Results

Fig. 2 shows the route establishment time of ISDSR-L and ISDSR+, in other words, the round trip time between a source node and its destination. The horizontal and vertical axes represent the number of hops and route establishment times, respectively. The results of ISDSR+ are faster than that of ISDSR-L. Both times increase in proportion to the number of hops. We consider that the packet size derives these results. In our implementation, the signature size of ISDSR+ and ISDSR-L is 240 bytes and 6176 bytes, respectively. Generally, the size of packets affects communication time. Naturally, ISDSR+ is faster than ISDSR-L because the signature size of ISDSR-L is more significant than that of ISDSR+.

TABLE I
SIGNATURE GENERATION AND VERIFICATION TIME

	Generation Time [ms]		Verification Time [ms]	
Hops	ISDSR-L	ISDSR+	ISDSR-L	ISDSR+
1	0.088	0.268	0.095	2.481
2	0.088	0.255	0.113	2.678
3	0.087	0.272	0.133	2.873
4	0.089	0.257	0.151	3.131
5	0.090	0.268	0.168	3.499
6	0.142	0.274	0.185	3.683
7	0.118	0.272	0.209	3.890
8	0.119	0.268	0.224	4.130
9	0.119	0.259	0.243	4.315

On the other hand, in terms of signature generation time and verification time, ISDSR-L is much faster than ISDSR+. Each result represents the average time over 100 runs in Table I. The column Hops in Table I represents the nth-hop node. For example, the results in the row "3" represents the signature generation time and verification time of 3rd-hop nodes. All signature generation times of ISDSR-L are less than half of ISDSR+. Regarding the signature verification, the results of ISDSR-L is much faster than that of ISDSR+.

Although the verification time of ISDSR-L is no more than 6% that of ISDSR+, the route establishment time of ISDSR-L is more than 300% in the nine-hop network. As mentioned before, it is evident that the signature size affects the route establishment time. We consider two ideas how to reduce the route establishment time. The first one is applying IEEE 802.11n or 802.11ac to shorten communication time. However, this idea depends on the device's specifications. The second is improving our aggregate signature scheme to suppress the signature size. The signature generated by our scheme consists of vectors. It is easy to suppress the signature size by reducing the bit length of each element in vectors. Even if the signature size is reduced by applying this idea, security level must be lower. We need to improve our aggregate signature scheme to suppress the signature size while maintaining security levels.

V. CONCLUSION

In this paper, we implemented ISDSR-L as a secure routing protocol for wireless multi-hop networks. To this end, we evaluated the route establishment time, signature generation time, and verification time of ISDSR-L compared with ISDSR+. From the point of view of the route establishment time, ISDSR-L needs more time than ISDSR+. One of the reasons ISDAR-L needs more time than ISDSR+ is that the signature size of ISDSR-L is more significant than that of ISDSR+. However, in terms of the signature generation time and verification time, ISDSR-L is much faster than ISDSR+.

There are several potential directions for future work. One of the future works is to make the route establishment time of

ISDSR-L faster by improving our aggregate signature scheme. The size of generated signatures in our aggregate signature scheme is more significant than that of ISDSR+ We plan to improve our aggregate signature scheme to suppress the signature size while maintaining security levels.

ACKNOWLEDGEMENT

This research was supported by JSJP KAKENHI 21K11866, 24K14937.

REFERENCES

[1] L. Zhou and Z. Haas, "Securing ad hoc network," *IEEE Network Magazine*, vol. 13, no. 6, pp. 24–30, 1999.

[2] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," *Ad Hoc Networks*, vol. 1, no. 2, pp. 293 – 315, 2003.

[3] Y.-C. Hu, A. Perrig, and D. Johnson, "Ariadne: a secure on demand routing protocol for ad hoc network," *Wireless Networks*, vol. 11, pp. 21–38, 2005.

[4] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks," *Wireless Networks*, vol. 11, no. 1-2, pp. 21–38, Jan. 2005.

[5] H. Kojima, N. Yanai, and J. P. Cruz, "Isdsr+: improving the security and availability of secure routing protocol," *IEEE Access*, vol. 7, pp. 74 849–74 868, 2019.

[6] M. Zapata and N. Asokan, "Securing ad hoc routing protocols," in *Proc. of WISE*. ACM Press, 2002, pp. 1–10.

[7] J. Kim and G. Tsudik, "Srdp: Secure route discovery for dynamic source routing in manets," *Ad Hoc Networks*, vol. 7, no. 6, pp. 1097–1109, 2009.

[8] U. Ghosh and R. Datta, "Identity based secure aodv and tcp for mobile ad hoc networks," in *Proc. of ACWR 2011*. ACM, 2011, pp. 339–346.

[9] K. Muranaka, N. Yanai, S. Okamura, and T. Fujiwara, "ISDSR: Secure DSR with ID-based Sequential Aggregate Signature," in *Proc.of ICETE 2016*, 2016, pp. 376–387.

[10] S. M. Muzammal, R. K. Murugesan, and N. Z. Jhanjhi, "A comprehensive review on secure routing in internet of things: Mitigation methods and trust-based approaches," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4186–4210, 2020.

[11] Y. Shibasaki, K. Iwamura, and K. Sato, "A communication-efficient secure routing protocol for iot networks," *Sensors*, vol. 22, no. 19, p. 7503, 2022.

[12] K. Itakura and K. Nakamura, "A public-key cryptosystem suitable for digital multi-signatures," *NEC Research and Development*, vol. 71, pp. 1–8, 1983.

[13] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *Proc. of EUROCRYPT 2003*, vol. 2656 of LNCS. Springer, 2003, pp. 416–432.

[14] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé, "Crystals-dilithium: A lattice-based digital signature scheme," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 238–268, 2018.

[15] C. Aguilar-Melchor, J. Barrier, S. Guelton, A. Guinet, M.-O. Killijian, and T. Lepoint, "Nfllib: Ntt-based fast lattice library," in *Topics in Cryptology-CT-RSA 2016: The Cryptographers' Track at the RSA Conference 2016*. Springer, 2016, pp. 341–356.

[16] S. Ookura, H. Kojima, and N. Yanai, "Performance evaluation of id-based aggregate signature scheme based on lattice for wireless multi-hop secure routing protocols," in *2024 7th International Conference on Information and Computer Technologies (ICICT)*. IEEE, 2024, pp. 462–467.

[17] Y.-C. Hu, A. Perrig, and D. Johnson, "Ariadne: a secure on demand routing protocol for ad hoc network," in *Proc. of MobiCom 2002*. ACM, 2002.

[18] K. Sanzgiri, D. LaFlamme, B. Dahill, B. N. Levine, C. Shields, and E. Belding-Royer, "Authenticated routing for ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 3, pp. 598–610, 2005.

[19] G. Acs, "Secure routing in multi-hop wireless networks," Ph.D. dissertation, Budapest University of Technology and Economics, 2009.

[20] U. Ghosh and R. Datta, "Sdrp: Secure and dynamic routing protocol for mobile ad-hoc networks," *IET Network*, vol. 3, no. 3, pp. 235–243, 2013.

[21] Y.-H. Lee, H. Kim, B. Chung, J. Lee, and H. Yoon, "On-demand secure routing protocol for ad hoc network using id based cryptosystem," in *Proc. of 4th ICPDCAT*. IEEE, 2003, pp. 211–215.

[22] J. Song, H. Kim, S. Lee, and H. Yoon, "Security enhancement in ad hoc network with id-based cryptosystem," in *Proc. of ICACT 2005*, vol. 1. IEEE, 2005, pp. 372–376.

[23] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. of CRYPTO 84*, ser. LNCS, vol. 196. Springer, 1984, pp. 47–53.

[24] R. El Bansarkhani and J. Buchmann, "Towards lattice based aggregate signatures," in *Progress in Cryptology–AFRICACRYPT 2014: 7th International Conference on Cryptology in Africa, Marrakesh, Morocco, May 28-30, 2014. Proceedings 7*. Springer, 2014, pp. 336–355.

[25] X. Lu, W. Yin, Q. Wen, Z. Jin, and W. Li, "A lattice-based unordered aggregate signature scheme based on the intersection method," *IEEE Access*, vol. 6, pp. 33 986–33 994, 2018.

[26] K. Boudgoust and A. Roux-Langlois, "Compressed linear aggregate signatures based on module lattices." *IACR Cryptol. ePrint Arch.*, vol. 2021, p. 263, 2021.

[27] Y. Quan, "Improving bitcoin's post-quantum transaction efficiency with a novel lattice-based aggregate signature scheme based on crystals-dilithium and a stark protocol," *IEEE Access*, vol. 10, pp. 132 472–132 482, 2022.

[28] S. Sato and J. Shikata, "Identity-based interactive aggregate signatures from lattices," in *International Conference on Information Security and Cryptology*. Springer, 2022, pp. 408–432.

[29] Q. Li, M. Luo, C. Hsu, L. Wang, and D. He, "A quantum secure and noninteractive identity-based aggregate signature protocol from lattices," *IEEE Systems Journal*, vol. 16, no. 3, pp. 4816–4826, 2022.

[30] A. Boldyreva, C. Gentry, A. O'Neill, and D. H. Yum, "Ordered multisignatures and identity-based sequential aggregate signatures, with applications to secure routing (extended abstract)," 2010. [Online]. Available: http://www.cc.gatech.edu/~aboldyre/papers/bgoy.pdf

[31] V. Lyubashevsky, "Fiat-shamir with aborts: Applications to lattice and factoring-based signatures," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2009, pp. 598–616.

[32] T. Güneysu, V. Lyubashevsky, and T. Pöppelmann, "Practical lattice-based cryptography: A signature scheme for embedded systems," in *Cryptographic Hardware and Embedded Systems–CHES 2012: 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings 14*. Springer, 2012, pp. 530–547.

[33] S. Bai and S. D. Galbraith, "An improved compression technique for signatures based on learning with errors," in *Topics in Cryptology–CT-RSA 2014: The Cryptographer's Track at the RSA Conference 2014, San Francisco, CA, USA, February 25-28, 2014. Proceedings*. Springer, 2014, pp. 28–47.

[34] R. R. Fontes, S.Afzal, S. H. B. Brito, M. A. S. Santos, and C. E. Rothenberg, "Mininet-wifi: Emulating software-defined wireless networks," in *Proc. 11th International Conference on Network and Service Management (CNSM)*, Nov. 2015, pp. 384–389.