ISDSR: Secure DSR with ID-based Sequential Aggregate Signature

Kenta Muranaka[1], Naoto Yanai[1], Shingo Okamura[2] and Toru Fujiwra[1]
[1]*Graduate School of Information Science and Technology, Osaka University, Osaka, Japan*
[2]*National Institute of Technology, Nara College, Nara, Japan*

Keywords: Secure Routing Protocol, Wireless Sensor Network, Dynamic Source Routing, Aggregate Signature, ID-based Signature, ID-based Sequential Aggregate Signature.

Abstract: Wireless sensor networks are often more vulnerable than wired ones. Especially, an adversary can attack the networks by utilizing false route information. A countermeasure against the attack is a secure routing protocol with digital signatures to guarantee the validity of route information. However, existing secure routing protocols are inefficient because the memory size and the computational overhead are heavy. To overcome these problems, we focus on ID-based sequential aggregate signatures (IBSAS) (Boldyreva et al., 2007). IBSAS allow users to aggregate individual signatures into a single signature. Moreover, certificates of public keys are unnecessary for IBSAS. Therefore, IBSAS can drastically decrease the memory size and the computational overhead. Besides, one of the main concerns for practical use is to construct a protocol specification with IBSAS. Moreover, since IBSAS are sometimes weak against compromising secret keys, another concern is to construct its countermeasure. For these purposes, we propose a secure dynamic source routing with ID-based sequential aggregate signatures, called ISDSR for short and discuss the key management to revoke/update compromised keys. We also show that the performance of ISDSR is the best in comparison with the existing protocols.

1 INTRODUCTION

1.1 Background

When devices send data in wireless sensor networks, intermediate devices relay the received data to its destination as routers. This mechanism is known as a multi-hop routing protocol. There are also two types of mechanisms; namely dynamic source routing (DSR) protocol (Johnson and Maltz, 1996) and an ad hoc on-demand distance vector (AODV) protocol (Perkins and Royer, 1999). Both protocols are well-known instantiations. From the standpoint of an adversary, wireless sensor networks are more vulnerable than wired ones since the adversary can attack by a setup of its own devices within the networks. For example, malicious devices can inject false route information (Karlof and Wagner, 2003), and then packets are looped or transferred unnecessarily to far nodes. A sinkhole attack (Karlof and Wagner, 2003) also exists, where data is collected to malicious nodes managed by an adversary.

One of the potential approaches to preventing these attacks is to guarantee the validity of the route information to secure routing protocols with authentication tools such as digital signatures. This has been proposed (Hu et al., 2002a; Zapata and Asokan, 2002). Several standardized organizations such as the European Telecommunications Standards Institute (ETSI) have suggested the use of digital signatures to prevent forgery of information or unauthorized accesses (Guillemin, 2007). We thus focus on digital signatures. Nevertheless, utilizing digital signatures brings problems for the memory size and the computational overhead. From another standpoint of sensor networks, the batteries of sensor devices are small and low-capacity. Hence, the memory size and the computational overhead should be small. Based on these mechanisms and approaches, our goal is to construct a secure routing protocol with both minimum memory size and lower computational overhead.

1.2 Problems for Secure Routing Protocols

The secure routing protocols discussed in this work are routing protocols with digital signatures that guar-

Muranaka, K., Yanai, N., Okamura, S. and Fujiwara, T.
ISDSR: Secure DSR with ID-based Sequential Aggregate Signature.
DOI: 10.5220/0006001003760387
In *Proceedings of the 13th International Joint Conference on e-Business and Telecommunications (ICETE 2016) - Volume 4: SECRYPT*, pages 376-387
ISBN: 978-989-758-196-0

antee the validity of the route information. In particular, each node generates signatures on route information when the node sends a packet including the route information. Any node that received the packets and the signatures verify its validity. However, a trivial use of digital signatures increases the data size of the packets linearly in proportion to the number of nodes. This then requires each node to contain a large size of memory storage and it also requires a large amount of computational time to verify all of the signatures. Moreover, since certificates for public keys are necessary, obtaining and verifying the certificates often require large cost. Although Kim and Tsudik (Kim and Tsudik, 2009), Ghosh and Datta (Ghosh and Datta, 2011) and Muranaka et al. (Muranaka et al., 2015) have proposed secure routing protocols to decrease the overhead of signatures by the use of multisignatures, they are still insufficient due to the verification of public keys via certificates. Verification including certificates for public key often require heavy cost, and thus the existing protocols are unpractical due to the large overhead.

1.3 Contributions

In this work, we propose a new secure routing protocol to guarantee the validity of the route information, which is the most efficient for the memory size and the computational overhead in existing secure routing protocols. In particular, in addition to combining individual signatures into a single short signature,verification of public keys via certificates are unnecessary for our protocol. Our main approach is to utilize ID-based sequential aggregate signatures (IBSAS) (Boldyreva et al., 2007). We name the proposed protocol a *secure DSR with ID-based sequential aggregate signatures (ISDSR).* The IBSAS use digital signatures to obtain both multisignatures (Itakura and Nakamura, 1983) where signatures are compressed into a single short signature and ID-based signatures (Shamir, 1984) so that any identity can be used as a public key for each user. Although there are several secure routing protocols (Kim and Tsudik, 2009; Muranaka et al., 2015) with multisignatures that decrease the size of signatures, they are still unpractical due to verifications of public keys via certificates. In comparison, our proposed protocol removes such verifications of public keys by virtue of ID-based signatures.

Our primary contribution is to propose a protocol specification of ISDSR for practical use. A new protocol specification is required for DSR with IBSAS because IBSAS are different from the conventional digital signatures in terms of multisignatures and ID-based signatures. Constructing an efficient and feasible specification is still a challenge problem. Furthermore, since identities are used as public keys and they are unique information, it is also difficult for ID-based signatures to revoke/update keys if the keys are compromised. Therefore, we clarify the specification and discuss the management of public keys. We show that the performance of ISDSR is the best relatively to the existing protocols. In this work, we focus on DSR as a routing protocol in ad hoc networks. We consider that the main construction of ISDSR can be extended to other routing protocols such as AODV.

2 RELATED WORKS

Zhou and Haas (Zhou and Haas, 1999) proposed the first secure routing protocol for key management on ad hoc networks. Hu et al. (Hu et al., 2002a; Hu et al., 2005) have proposed Ariadne in DSR (Johnson and Maltz, 1996) with message authentication codes (MAC) and protocols with digital signatures. Next, Papadimitratos and Haas (Papadimitratos and Haas, 2002) have proposed another protocol called secured routing protocol (SRP) with symmetric cryptography. In comparison with Ariadne, SRP makes no assumptions regarding intermediate nodes between a source and a destination.Hu et al. have also proposed SEAD (Hu et al., 2002b) in destination-sequenced distance vector protocol (DSDV) (Broch et al., 1998). SEAD has utilized MAC. In AODV (Perkins and Royer, 1999), Zapata and Asokan firstly proposed secure AODV (SAODV) (Zapata and Asokan, 2002). SAODV is based on both types of cryptography. Following these authors, Sangiri et al. (Sanzgiri et al., 2005) showed the vulnerabilities of SAODV and the proposed ARAN to improve the vulnerabilities by utilizing public key cryptography.

In recent years, there are three protocols with signature schemes for multiple signers. The first protocol is the secure route discovery protocol (SRDP) by Kim and Tsudik (Kim and Tsudik, 2009), and they have utilized sequential aggregate signatures (Lysyanskaya et al., 2004) and accountable-subgroup multisignatures (Boldyreva, 2003; Micali et al., 2001). Ghosh and Datta (Ghosh and Datta, 2011) have proposed identity based secure AODV (IDSAODV) with the sequential aggregate signatures. IDSAODV is the second protocol. However, we mention that their meaning “identity-based” is different from the use of ID-based signatures. That is, IDSAODV does not utilize ID-based signatures, but the sequential aggregate signature schemes. Moreover, public keys of all nodes for IDSAODV are registered in each node in

advance. Thus, IDSAODV is impractical. The third protocol is the secure routing protocols by Muranaka et al. (Muranaka et al., 2015), and these protocols are based on ordered multisignatures (Boldyreva et al., 2007; Yanai et al., 2013) and general aggregate signatures (Boneh et al., 2003). However, they are still insufficient as described in the previous section.

In the ID-based setting, Ghosh and Datta (Ghosh and Datta, 2013) also have proposed the secure dynamic routing protocol (SDRP) utilizing an ID-based key exchange protocol. However, the cost of their work seems to be inefficient compared to the works of (Kim and Tsudik, 2009; Muranaka et al., 2015) since SDRP requires each node to keep the linear size of shared keys with respect to the number of signers. Namely, each node has to keep both ID information for other nodes and its shared keys to communicate each other to verify the validity of route information. We emphasize that it is quite different from our work in the sense of the use of ID-based signature schemes for multiple signers. More precisely, each node for our protocol can keep only ID information to verify validity.

3 DYNAMIC SOURCE ROUTING

DSR (Johnson and Maltz, 1996) is a routing protocol designed for an ad hoc network. It can construct autonomous sensor networks. DSR has two functions, i.e., *route discovery* and *route maintenance*. The route discovery searches for a route and constructs a route when a transfer is required. The route maintenance restores a route which has broken.

3.1 Route Discovery

The route discovery finds a route when a node has a packet to send to a destination. The route discovery consists of a search phase with *route request packets* (RREQs) and a reply phase with *route reply packets* (RREPs).

In the search phase, a source node broadcasts RREQs around the node. Intermediate nodes write route information to the RREQ by adding their own IDs to the RREQ and broadcasting this information to neighbor nodes. This process is iterated until a destination node receives the RREQ. Then the destination node knows the route information through the RREQ. Since RREQ is transmitted as a local broadcast, a node often receives the RREQ transmitted by the node-self in advance. In such a case, the node checks whether its own ID is written in RREQ. If so, the node detects a loop and discards the RREQ. The RREQ consists of header and route information as in Figure 1. According to RFC4728 (Johnson et al., 2007), a source node ID and a destination node ID are included in the IP field. However, we do not assume the other layers, and consider that these IDs are included in the route information.

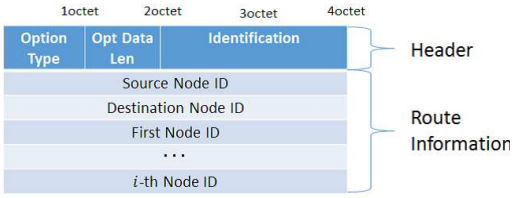

Figure 1: Route request packet (RREQ).

In the reply phase, the destination node sends the RREP to the source node when it receives the RREQ. The RREP is given the route which has been found by the RREQ. The destination node informs the source node of the route by sending the RREP via the reverse order of the route. When the source node receives the RREP, the route between the source node and the destination node is constructed. Then the source node sends data to the destination node via the route. The source node stores the route information in the cache and reuses it when it sends data to the same destination node.

3.2 Route Maintenance

In Route Maintenance, when a node detects a broken route, it sends other nodes *route error packets* (RERRs) and informs them of the destruction of the route. The node which has detected the broken route, for a short detect node, checks its own cache to see whether there are substitute routes. If so, the detect node changes the route to the substitute route and continues to send data via the substitute route. Otherwise, the detect node sends a RERR to the source node which has sent data via the broken route. If the nodes which received the RERR have the broken route in their cache, they delete the route in their cache. RERR consists of a header and error information as described in Figure 2. In the route maintenance, 3 is set in error type and the broken node is described in type specific information. Therefore, RERR is constructed as in Figure 3 in the route main-

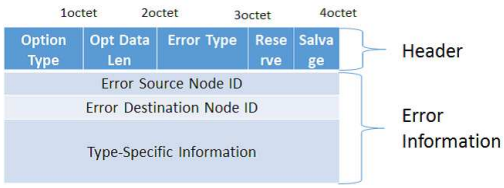

Figure 2: Route error packet (RERR).

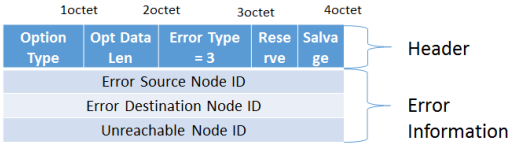

Figure 3: Route error packet (RERR) for route maintenance.

tenance. That is, the detect node ID is described in the Error Source Node ID and the source node ID which has sent data via the broken route is described in the Error Destination ID. The broken node ID is then described in the Unreachable Node ID.

4 ISDSR Model

4.1 Security Requirements

Sanzgiri et al. (Sanzgiri et al., 2005) argued that secure routing protocols should satisfy five security requirements when a route is constructed. However, these requirements can be integrated into two requirements. Therefore, we designed ISDSR to satisfy the following two requirements:.

1. Attackers cannot spoof other nodes.
2. Attackers cannot forge route information.

We define "guaranteeing the validity of the route information" as satisfying the above two requirements. ISDSR guarantees the validity of route information.

4.2 Assumptions

Node Assumptions. The network consists of two kinds of nodes, i.e., sensor nodes and a management server. We hereafter refer to sensor nodes as nodes. The management server manages other nodes. In other words, the management server issues an identity for each node, and generates a secret key for each node. We explain this in more detail in Section 6.1.

Network Assumptions. In the network, data is transmitted in both directions.

4.3 Attack Model

We define honest nodes as nodes which are managed by the management server, and attacker nodes as nodes which are non-managed. Honest nodes work as a specification of the protocol. The purpose of attackers is to attack networks which consist of only honest nodes to set their own nodes. This network is opened and the attacker can eavesdrop and manipulate data. Although sensor networks are vulnerable to DoS attacks such as jamming on the physical layer, we do not consider these attacks in this paper. We also do not consider the attacks on a media access control protocol. The attackers cannot hijack, rob, or destroy honest nodes. Attackers forge and manipulate the route information so that packets go through attack nodes. This attack model is equivalent to the definition of that a " managed-open attack" according to Sanzgiri et al. (Sanzgiri et al., 2005).

5 DESIGN PRINCIPLE

5.1 Foundation of ID-based Signature Schemes for Multiple Signers

As ID-based signature schemes for multiple signers can aggregate signatures into a single signature, the size of the signatures is constant with respect to the number of signers. Because they are ID-based signatures, certificates of public keys are unnecessary. Currently, there are ID-based aggregate signatures (IBAS) (Gentry and Ramzan, 2006) and ID-based sequential aggregate signatures (IBSAS) (Boldyreva et al., 2007) as ID-based signature schemes for multiple signers. The difference between these two schemes is the timing of the aggregation of the signatures. In IBAS, any signer can aggregate individually generated signatures. On the other hand, each signer for IBSAS generates signatures and their aggregation on the same algorithm. The restriction of the timing of aggregation in IBSAS is stronger than that in IBAS. However, each signer can only send a single signature in IBSAS. The functionality of IBSAS is sufficient for exchanging one signature between nodes in a secure routing protocol. The other difference is the necessity of a time stamp. It is necessary for IBAS and it is unnecessary in IBSAS. Since DSR does not require the time stamp, the use of IBSAS is more suitable than IBAS. Based on these observations, we utilize IBSAS in DSR.

5.2 Identity-based Sequential Aggregate Signatures

5.2.1 Algebraic Setting

Let $\mathbb{G}, \mathbb{G}_{\mathbb{T}}$ be groups of the same prime order p. A *paring* is an efficiently computable map $\mathbf{e}$: $\mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_{\mathbb{T}}$ such that the following two conditions hold;

- Bilinearity: For all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}$, we have $\mathbf{e}(au, bv) = \mathbf{e}(u, v)^{ab}$.
- Non-degeneracy: For any generator $g \in \mathbb{G}^*$, we have $\mathbf{e}(g, g) \neq 1_{\mathbb{G}_T}$ where $1_{\mathbb{G}_T}$ is an identity element over $\mathbb{G}_T$. That is $\mathbf{e}(g, g)$ generates $\mathbb{G}_T$

We call $(p, \mathbb{G}, \mathbb{G}_T, \mathbf{e})$ a bilinear-map parameter.

5.2.2 Algorithm

IBSAS have four algorithms, i.e., **Setup**, **KeyDerivation**, **Signing** and **Verification**. These algorithms are described in **Algorithm** 1-4.

Algorithm 1: Setup.

Ensure: Master public key mpk and master secret key msk

1: generate the bilinear-map parameter $(p, \mathbb{G}, \mathbb{G}_T, \mathbf{e})$
2: $g_1, g_2 \xleftarrow{R} \mathbb{G}$
3: $\alpha_1, \alpha_2 \xleftarrow{R} \mathbb{Z}_p$
4: $\{\mathbf{H}_1\}, \{\mathbf{H}_2\} : \{0,1\}^* \to \mathbb{G}$
5: $\{\mathbf{H}_3\} : \{0,1\}^* \to \mathbb{Z}_p$
6: **return** $(\mathbb{G}, \mathbb{G}_T, \mathbf{e}, g, \alpha_1 g, \alpha_2 g, \{\mathbf{H}_i\}_{i=1,\ldots,3})$ as the mpk and (α_1, α_2) as the msk

Algorithm 2: KeyDerivation.

Require: $msk, ID \in \{0,1\}^*$
Ensure: ID's secret key sk_{ID}

1: **return** $(\alpha_1 \mathbf{H}_1(ID), \alpha_2 \mathbf{H}_2(ID))$

Algorithm 3: Signing.

Require: $sk_{ID_i}, m \in \{0,1\}^*, \sigma, L_{i-1} = ((ID_1, m_1), \ldots, (ID_{i-1}, m_{i-1})) \in \{\{0,1\}^* \times \{0,1\}^*\}^{i-1}$
Ensure: signature $\sigma' = (\sigma'_1, \sigma'_2, \sigma'_3)$

1: **if** $i = 1$ **then**
2: σ is defined as $(1_{\mathbb{G}}, 1_{\mathbb{G}}, 1_{\mathbb{G}})$
3: **end if**
4: parse σ as $(\sigma_1, \sigma_2, \sigma_3)$
5: $r, x \xleftarrow{R} \mathbb{Z}_p$
6: $\sigma'_3 \leftarrow \sigma_3 + xg$
7: $\sigma'_2 \leftarrow \sigma_2 + rg$
8: $\sigma'_1 \leftarrow \sigma_1 + r\sigma_3 + x\sigma'_2 + \alpha_2 \mathbf{H}_2(ID_i) + \mathbf{H}_3(ID_i || m_i)\alpha_1 \mathbf{H}_1(ID_i)$
9: **return** $(\sigma'_1, \sigma'_2, \sigma'_3)$ as the ID's signature

Algorithm 4: Verification.

Require: $mpk, ((ID_1, m_1), \ldots, (ID_n, m_n)), \sigma$
Ensure: 1 or 0

1: **if** all of $ID_1, \ldots, ID_n$ are distinct **then**
2: parse σ as $(\sigma_1, \sigma_2, \sigma_3)$
3: **if** $\mathbf{e}(\sigma_1, g) \stackrel{?}{=} \mathbf{e}(\sigma_2, \sigma_3) \cdot \mathbf{e}(\prod_{i=1}^{n} \mathbf{H}_3(ID_i || m) \mathbf{H}_1(ID_i), \alpha_1 g)$ **then**
4: **return** 1
5: **else**
6: **return** 0
7: **end if**
8: **else**
9: **return** 0
10: **end if**

6 CONSTRUCTION OF ISDSR

ISDSR consists of four functions, i.e., *setup, secure route discovery, secure route maintenance* and *key management.* Notations in this section are as described in Table 1. In this work, we assume that headers are updated in a similar manner to that of the conventional DSR. In other words, we leave this as an open problem where we discuss the detail of the headers.

Table 1: Notations and their descriptions.

Notation	Description
ϕ	empty set
$ID_a \| \| ID_b$	concatenation of identities ID_a and ID_b,
ID_s	source node ID
ID_d	destination node ID
ID_i	i-th node ID
ID_c	detect node ID
ID_e	broken node ID
ID_r	revoked ID
ID_u	updating node ID
ID_{MS}	management server ID
msk	master secret key
mpk	master public key
sk_{ID_i}	ID_i's secret key
σ_{ID_i}	signature generated by ID_i in **Signing**
hd	header of packet
RI	route information
$SRREQ_{ID_i}$	$SRREQ$ generated by ID_i
EI	error information
ET	error type number
k_{ID}	shared key between the management server and ID
$Enc(k, m)$	ciphertext for a message m by a shared key k
$Dec(k, c)$	message decoded for a ciphertext c by a shared key k

6.1 Setup

A management server generates a master public key and a master secret key through the **Setup** of IBSAS.

The management server issues an identity for each of the nodes, and generates an ID's secret key corresponding to the ID for each of the nodes through the **KeyDerivation** of IBSAS. The management server also generates a shared key for each of the nodes to communicate encrypted messages between the management server and each of the nodes. Each of nodes pre-installs the master public key, the secret key for the node's IDs and the shared key.

6.2 Secure Route Discovery

6.2.1 Basic Concept

ISDSR constructs a route via the secure route discovery instead of the route discovery in DSR, and guarantees that there are no attacker nodes in the route. ISDSR searches a route with *secure route request* packets (SRREQs), and construct the route with *secure route reply* packets (SRREPs).

6.2.2 Search Phase

The search phase for the secure route discovery is the same as that in DSR except for adding each node's signature to a packet. Namely, SRREQ consists of a header, route information and a signature as described in Figure 4. A source node writes its own ID in route information. The source node then generates a signature for route information through the **Signing** of IBSAS and adds the signature to the SRREQ. The source node broadcasts the SRREQ to neighbor nodes. A node that receives the SRREQ adds its own ID to the route information. The node generates a new signature for the route information as a message and the signature in SRREQ by **Signing**. The node replaces the old signature with the new signature in the SRREQ, and broadcasts it around the node. When the destination node receives the SRREQ, it verifies the signature in the SRREQ using received intermediate nodes IDs by **Verification** of the IBSAS. If the signature is valid, the destination node starts the reply phase. Otherwise, the node waits until the node receives the SRREQ whose signature is valid.

6.2.3 Reply Phase

In the reply phase, the destination node generates a signature for route information by **Signing**. It adds the signature to SRREP and sends the SRREP to its source node. When the source node receives the SRREP, the source node verifies the signature in the SRREP **Verification** with the destination node ID. If the signature is valid, the route is constructed. Otherwise, the source node discards the SRREP.

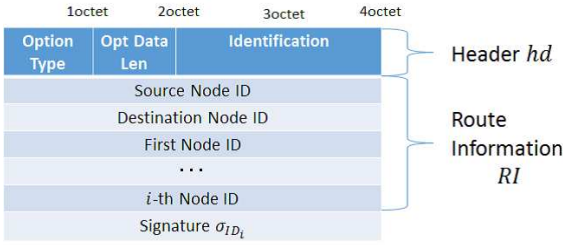

Figure 4: Secure request packet for i-th node ($SRREQ_{ID_i}$).

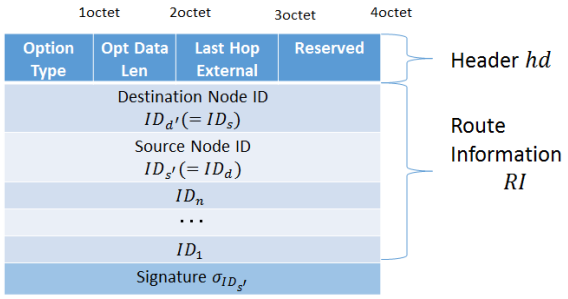

Figure 5: Secure route reply packet (SRREP).

6.2.4 Algorithms

Notations are as described in Table 1. SRREQ consists of three elements as described in Figure 4; i.e., a header *hd*, route information *RI* and a signature σ. SRREP consists of three elements as described in Figure 5; i.e., a header *hd*, route information *RI* and a signature σ. A header in ISDSR is the same as that in DSR. *RI* consists of three kinds of identity; i.e., a source node ID, a destination node ID and the intermediate nodes ID. A destination node in SRREP is the same as a source node in SRREQ. *RI* in SRREP is given as the reverse order of *RI* in SRREQ. Functions **SRREQConstruct**C**SRREPConstruct**, **LoopCheck** and **SRREPHeaderUpdate** are described in **Function** 1 - 4D**SRREQConstruct** which takes as input a header, route information and a signature, is a function which updates the header in a similar manner as DSR, and constructs a SRREQ with these three kinds of elements, i.e., the header, the route information and the signature. **LoopCheck** which takes as input a node ID and route information, is a function which checks whether the node ID is included in the route information. If so, it returns 1. Otherwise, it returns 0. **SRREPConstruct** which takes as input route information and a signature, is a function which constructs the header in a similar manner as that of the conventional DSR, and a SRREP with the header and the route information. The route information in SRREP is the reverse order of route information described in the input in a similar manner as that of the conventional DSR. **SRREPHeaderUpdate** which takes as input SRREP, is a function which constructs the SR-

REP, in which the header is updated. **Algorithm** 5 - 10 are algorithms in the secure route discovery.

Function 1: SRREQConstruct.

Require: RI, σ, hd
Ensure: $SRREQ = (hd, RI, \sigma)$
1: update hd
2: $SRREQ \leftarrow (hd, RI, \sigma)$
3: **return** $SRREQ$

Function 2: LoopCheck.

Require: RI, ID
Ensure: 1 or 0
1: **if** $ID \in RI$ **then**
2: **return** 1
3: **else**
4: **return** 0
5: **end if**

Function 3: SRREPConstruct.

Require: RI', σ
Ensure: $SRREP = (hd, RI, \sigma)$
1: construct a header hd.
2: $RI \leftarrow$ reverse order of RI'
3: $SRREP \leftarrow (hd, RI, \sigma)$
4: **return** $SRREP$

Function 4: SRREPHeaderUpdate.

Require: $SRREP = (hd', RI, \sigma)$
Ensure: $SRREP = (hd, RI, \sigma)$
1: update the header
2: **return** $SRREP$

Algorithm 5: The search phase on a source node.

Require: sk_{ID_s}, ID_s, ID_d
Ensure: $SSREQ_{ID_s} = (hd_0, RI_0, \sigma_{ID_s})$
1: $RI_0 \leftarrow \phi$
2: $hd \leftarrow \phi$
3: $RI_0 \leftarrow RI_0 || ID_d || ID_s$
4: $\sigma_{ID_s} \leftarrow$ **Signing**$(sk_{ID_s}, RI_0, 1)$
5: $SSREQ_{ID_s} \leftarrow$ **SRREQConstruct**(hd, RI_s, σ_s)
6: **return** $SSREQ_{ID_s}$

6.3 Secure Route Maintenance

6.3.1 Basic Concept

A node detects a link break to the next hop in data transfer. The detect node sends a *secure route error packets* (SRERRs) to a source node. The SRERR is as described in Figure 6, and includes a signature generated by the detect node via **Signing**. Intermediate nodes between the detect node and its source node just transfer the SRERR. The source node verifies the signature by **Verification** with the detect node ID and transfers the data again by a route which does not include the broken link.

Algorithm 6: The search phase on an i-th node.

Require: ID_i, sk_{ID_i},
$SRREQ_{ID_{i-1}} = (hd_{i-1}, RI_{i-1}, \sigma_{ID_{i-1}})$
Ensure: $SSREQ_{ID_i} = (hd_i, RI_i, \sigma_{ID_i})$ or 0
1: **if** **LoopCheck**$(RI_{i-1}, ID_i) = 1$ **then**
2: discard $SRREQ_{ID_{i-1}}$
3: **return** 0
4: **else**
5: $RI_i \leftarrow RI_{i-1} || ID_i$
6: $\sigma_{ID_i} \leftarrow$ **Signing**$(sk_{ID_i}, RI_i, \sigma_{ID_{i-1}})$
7: $SSREQ_{ID_i}$
$\leftarrow$ **SRREQConstruct**$(hd_{i-1}, RI_i, \sigma_{ID_i})$
8: **return** $SSREQ_{ID_i}$
9: **end if**

Algorithm 7: The search phase on a destination node.

Require: $SRREQ_{ID_n} = (hd_n, RI_n, \sigma_{ID_n})$
Ensure: 1
1: **while** **Verification**$(mpk, RI_n, \sigma_{ID_n}) \neq 1$ **do**
2: wait other $SRREQ$s
3: **end while**
4: **return** 1

Algorithm 8: The reply phase on a destination node.

Require: sk_{ID_d}, $SRREQ_{ID_n} = (hd_n, RI_n, \sigma_{ID_n})$
Ensure: $SRREP = (hd, RI, \sigma_{ID_d})$
1: $\sigma_{ID_d} \leftarrow$ **Signing**$(sk_{ID_d}, RI_n, 1)$
2: $SRREP \leftarrow$ **SRREPConstruct**(RI_n, σ)
3: **return** $SRREP$

Algorithm 9: The reply phase on a i-th node.

Require: $SRREP = (hd', RI, \sigma_{ID_d})$
Ensure: $SRREP = (hd, RI, \sigma_{ID_d})$
1: $SRREP \leftarrow$ **SRREPHeaderUpdate**$(SRREP)$
2: **return** $SRREP$

6.3.2 Algorithms

Algorithms 11 and 12 are algorithms for secure route maintenance. SRERR is as described in Figure 6. SRERR in the secure route maintenance is the same

Algorithm 10: The reply phase on a source node.

Require: $SRREP = (hd, RI, \sigma_{ID_d})$
Ensure: 0 or 1
1: **if** **Verification**$(mpk, \sigma_{ID_d}, RI) = 1$ **then**
2: construct a route
3: **return** 1
4: **else**
5: discard $SRREP$
6: **return** 0
7: **end if**

as RERR in the route maintenance of DSR except for the addition of a signature. Function **SRERRConstruct** is described in **Function** 5D**SRERRConstruct** which takes as input error information EI, a signature σ and the number ET is a function which constructs SRERR with error type ET. **Algorithms**11 and 12 are algorithms in the secure route maintenance.

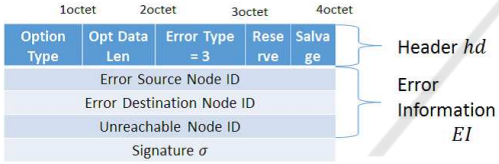

Figure 6: Secure route error packet (SRERR) for secure route maintenance.

Function 5: SRERRConstruct.
Require: EI, σ, ET
Ensure: $SRERR$
1: construct a header hd which Error Type$= ET$.
2: $SRERR \leftarrow (hd, EI, \sigma)$
3: **return** $SRERR$

Algorithm 11: A detect node.

Require: $sk_{ID_c}, ID_c, ID_s, ID_e$
Ensure: $SRERR$
1: $EI \leftarrow \phi$
2: $EI \leftarrow EI||ID_c||ID_s||ID_e$
3: $\sigma_{ID_d} \leftarrow$ **Signing**$(sk_{ID_c}, EI, 1)$
4: $SRERR \leftarrow$ **SRERRConstruct**$(EI, \sigma_{ID_c}, 3)$
5: **return** $SRERR$

6.4 Key Management

Key management is a mechanism to revoke and update the ID. The key management utilizes SRERR extended from RERR. The SRERR has two kinds of error types, i.e., a key revocation type (Error Type=4) and key update type (Error Type=5) for key management.

Algorithm 12: A source node.

Require: $SRERR = (hd, EI, \sigma_{ID_c})$
Ensure: 1 or 0
1: **if** **Verification**$(mpk, \sigma_{ID_c}, ID_c) = 1$ **then**
2: confirm the broken link.
3: **return** 1
4: **else**
5: discard the $SRERR$.
6: **return** 0
7: **end if**

6.4.1 Key Revocation

Basic Concept. When a node ID is revoked as a public key, a management server broadcasts a SRERR included in the node ID and a signature generated by the management server via **Signing** to other nodes. When nodes receive the SRERR, they verifies the signature in the SRERR by **Verification** with the management server ID. If it is valid, nodes delete routes included in the revocation ID in their caches. They store the revocation ID in their caches to check whether the ID is in routes under route construction.

Algorithms. **Algorithms** 13 and 14 are algorithms in Key Revocation. The SRERR in Key Revocation is as described in Figure 7. The Error Type is 4, the Error Source Node ID is the management server ID, the Error Destination Node ID is broadcast and Type-Specific Information is the revocation ID in the SRERR. The signature generated by the management server is added to the SRERR.

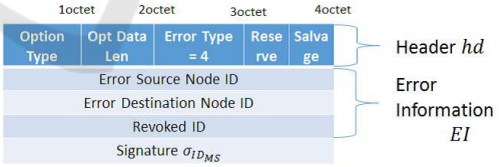

Figure 7: Secure route error packet (SRERR) for key revocation.

Algorithm 13: A management server.

Require: $ID_{MS}, ID_r, sk_{ID_{MS}}$
Ensure: $SRERR$
1: $EI \leftarrow \phi$
2: $EI \leftarrow EI||ID_{MS}||Broadcast||ID_r$
3: $\sigma_{ID_{MS}} \leftarrow$ **Signing**$(sk_{ID_{MS}}, EI, 1)$
4: $SRERR \leftarrow$ **SRERRConstruct**$(EI, \sigma_{ID_{MS}}, 4)$
5: **return** $SRERR$

Algorithm 14: Nodes.

Require: $SRERR = (hd, EI, \sigma_{ID_{MS}})$
Ensure: $SRERR$ or 0
1: **if** **Verification**$(mpk, \sigma, ID_{MS}) = 1$ **then**
2: delete routes included in the ID in their caches.
3: store ID_r in their caches as the revocation ID.
4: broadcast $SRERR$.
5: **return** $SRERR$
6: **else**
7: discard $SRERR$.
8: **return** 0
9: **end if**

6.4.2 Key Update

Basic Concept. A management server issues a new ID and generates a secret key corresponding to the ID in **KeyDerivation**. The management server sends a SRERR included in the new ID, the secret key and a signature generated by the management server via **Signing** to the node in which ID is updated. The secret key is encrypted by a shared key between the management server and the node. The node verifies the signature in SRERR by **Verification** with the management server ID. If the signature is valid, the node updates its ID and stores the secret key decoded by the shared key. Then the management server revokes the old ID with a SRERR in the key revocation.

Algorithms. **Algorithms** 15 and 16 are algorithms in the key update. The SRERR of the key update is described in Figure 8. The Error Type is 5, the Error Source Node ID is the management server ID and the Error Destination Node ID is the updating node ID. A new node ID and an encrypted secret key is described in the Type-Specific Information. A signature generated by the management server is added to the SRERR.

Algorithm 15: A management server.

Require: ID_{MS}, $sk_{ID_{MS}}$
Ensure: $SRERR$
1: $EI \leftarrow \phi$
2: $EI \leftarrow EI||ID_{MS}||ID_u||ID_{new}||Enc(k_{ID_u}, sk_{ID_{New}})$
3: $\sigma_{ID_{MS}} \leftarrow$ **Signing**$(sk_{ID_{MS}}, EI, 1)$
4: $SRERR \leftarrow$ **SRERRConstruct**$(EI, \sigma_{ID_{MS}}, 5)$
5: revoke ID_u in **Key Revocation**
6: **return** $SRERR$

Algorithm 16: An updating node.

Require: $SRERR = (hd, EI, \sigma_{ID_{MS}})$
Ensure: 1 or 0
1: **if** **Verification**$(mpk, \sigma_{ID_{MS}}, ID_{MS}) = 1$ **then**
2: $ID_u \leftarrow ID_{New}$
3: $sk_u \leftarrow Dec(k_{ID_u}, Enc(k_{ID_u}, sk_{ID_{New}}))$
4: store ID_u as revocation node ID in its cache.
5: **return** 1
6: **else**
7: discard $SRERR$.
8: **return** 0
9: **end if**

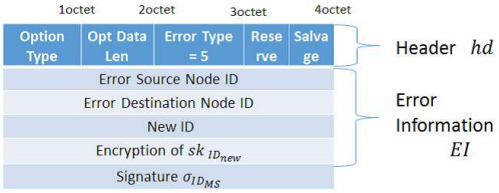

Figure 8: Secure route error packet (SRERR) for key update.

7 DISCUSSION

7.1 Performance Analysis

We show IDSDR can decrease the memory size and the computational overhead. That is, *routing overhead* (RO) and *routing latency* (RL) of ISDSR are smaller than those of other secure routing protocols. These are main bottlenecks for secure routing protocols. Thus, these decreasement is important for prati-

Table 2: Notations and their descriptions in numerical analysis.

Notation	Description
ID_{SIZE}	size of node ID
$RSASig_{SIZE}$	size of RSA-based signature (2048bit)
$RSASig_{GEN}$	RSA-based signature generation time
$RSASig_{VER}$	RSA-based signature verification time
$ECCSig_{SIZE}$	size of ECC-based signature (224bit)
$ECCSig_{GEN}$	ECC-based signature generation time
$ECCSig_{VER}$	ECC-based signature verification time
$Cert_{SIZE}$	size of public key certificate(about 1KB)
$Cert_{VER}$	certificate verification time
MAC_{SIZE}	size of MAC (128bit)
MAC_{GEN}	MAC generation time
MAC_{VER}	MAC verification time
KEY_{GEN}	shared key generation time
$Hash_{SIZE}$	size of hash (128bit)
t	number of intermediate nodes

cal use. Table 2 shows notations and their descriptions in this section.

7.1.1 Routing Overhead (RO)

The RO here refers to the total number of bytes that are transmitted over the network to establish a secure path from any source to any destination. As DSR does not include any security mechanism, the RO of the DSR with t intermediate nodes to establish a valid route is expressed by the following equation:

$$RO_{DSR} = (t+1) \times (RREQ_{SIZE} + RREP_{SIZE}). \quad (1)$$

We consider the RO of ISDSR. In ISDSR, each intermediate node sends a single signature to the next node in both the search phase and the reply phase.

Therefore the RO of the ISDSR with t intermediate nodes is computed by the following equation:

$$RO_{ISDSR} = RO_{DSR} + 2 \times (1+t) \times ECCSIG_{SIZE}. \quad (2)$$

Ghosh and Datta (Ghosh and Datta, 2013) calculate ROs of other secure routing protocols as follows:

$$\begin{aligned} RO_{ARAN} =& RO_{AODV} + 2 \times (1+2 \times t) \\ & \times (RSASig_{SIZE} + Cert_{SIZE}), \end{aligned} \quad (3)$$

$$\begin{aligned} RO_{SAODV} =& RO_{AODV} + 2 \times (1+t) \\ & \times (RSASig_{SIZE} + Cert_{SIZE} + Hash_{SIZE}), \end{aligned} \quad (4)$$

$$\begin{aligned} RO_{IDSAODV} =& RO_{AODV} \\ & + 2 \times (1+t) \times ECCSig_{SIZE} \\ & + (3/2 \times t^2 + t/2 + 2) \times HASH_{SIZE}, \end{aligned} \quad (5)$$

$$\begin{aligned} RO_{SDRP} =& RO_{AODV} + (2+3 \times t) \times ECCSig_{SIZE} \\ & + t \times MAC_{SIZE} \\ & + (3/2 \times t^2 + t/2 + 2) \times ID_{SIZE}. \end{aligned} \quad (6)$$

Ghosh and Datta (Ghosh and Datta, 2013) have calculated $RO_{IDSAODV}$ by using ID_{SIZE} instead of $HASH_{SIZE}$. However, IDSAODV needs the hash values of its public keys of RSA as ID. Therefore, we calculate $RO_{IDSAODV}$ by using $HASH_{SIZE}$. SDRP needs hardware addresses of nodes as IDs in addition to IDs of nodes, which are used in the original specification of DSR. We denote by ID_{SIZE} the overhead based on this redundant information. However, such redundant information is unnecessary for ISDSR because the overhead with respect to the IDs in ISDSR is identical to RO_{DSR}.

DSR is extremely similar to AODV. Therefore it is fair to consider $RO_{DSR} = RO_{AODV}$. The RO of DSR is smaller than that of other secure routing protocols.

7.1.2 Routing Latency (RL)

RL of a routing protocol is the time required to establish a valid secure route. In the case of a secure routing protocol, RL includes latency because of route discovery and latency because of the addition of security mechanisms in the protocol.

In Secure Routing Discovery, a signature is generated $(1+t)$ times and the signature is verified only one time in a destination node.

$$\begin{aligned} RL_{ISDSR} =& RL_{DSR} + (2+t) \times ECCSig_{GEN} \\ & + 2 \times ECCSig_{VER}. \end{aligned} \quad (7)$$

Ghosh and Datta (Ghosh and Datta, 2013) also calculate the RLs of other secure routing protocols as follows:

$$\begin{aligned} RL_{ARAN} =& RL_{AODV} + 2 \times (1+t) \times RSASig_{GEN} \\ & + 2 \times (2+t) \times RSASig_{VER} \\ & + 2 \times (2+t) \times Cert_{VER}, \end{aligned} \quad (8)$$

$$\begin{aligned} RL_{SAODV} =& RL_{AODV} + 2 \times (1+t) RSASig_{GEN} \\ & + 2 \times (1+t) \times RSASig_{VER} \\ & + 2 \times (1+t) \times Cert_{VER}, \end{aligned} \quad (9)$$

$$\begin{aligned} RL_{IDSAODV} =& RL_{AODV} + 2 \times (1+t) \times RSASig_{GEN} \\ & + 3 \times (1+t) \times RSASig_{VER}, \end{aligned} \quad (10)$$

$$\begin{aligned} RL_{SDRP} =& RL_{AODV} + (2+t) \times ECCSig_{GEN} \\ & + (3+2 \times t) \times ECCSig_{VER} \\ & + t \times (MAC_{GEN} + MAC_{VER}) \\ & + (1+t) \times Key_{GEN}. \end{aligned} \quad (11)$$

The RL of the DSR is also the same as that of the AODV. We know that computing ECC-based signatures is faster than computing RSA-based signatures. The shared generation involves bilinear pairing, which is also fast. Therefore the RL of the ISDSR is comparable with any of the existing routing protocols.

7.2 Security Analysis

If route information is forged or there are attacker nodes in the route information, a signature for the route information is rejected by verification. In other words, if a route is constructed with a valid signature, the route consists of only honest nodes. That is, ISDSR satisfies requirement 1 in Section 4.1. Because we do not consider that attackers rob nodes and their control in ISDSR, the attacker cannot know any

node's secret key. Therefore, the attackers cannot forge signatures. That is, ISDSR satisfies requirement 2 in Section 4.1. Based on these observations, ISDSR can guarantee the validity of the route information when a route is constructed. Meanwhile, the security level of ISDSR is equivalent to that of ARAN, i.e., "managed-open".

8 CONCLUSION

We proposed ISDSR extended from DSR, which is the routing protocol in sensor networks, and we considered ISDSR's security for the importance of the security of sensor networks. We showed that ISDSR guarantees that there are no attacker nodes in the constructed routes. Our future research considers further implementation and evaluation of ISDSR. We focus on the use of NS-3[1] to evaluate ISDSR. Other future research will consider stricter security analysis. In particular, we plan to use formal methods as described in (Arnaud et al., 2010; Arnaud et al., 2014; Zhang et al., 2014). We also plan to extend other routing protocols in ad hoc networks, e.g, AODV, to utilize IBSAS in the future.

ACKNOWLEDGEMENTS

This research was supported in part by the Japan Society for the Promotion of Science KAKENHI Number 16K16065. We appreciate their support.

REFERENCES

Arnaud, M., Cortier, V., and Delaune, S. (2010). Modeling and verifying ad hoc routing protocols. In *Proc. of CSF 2010*, pages 59–74. IEEE.

Arnaud, M., Cortier, V., and Delaune, S. (2014). Modeling and verifying ad hoc routing protocols. *Information and Computation*, 238:30–67.

Boldyreva, A. (2003). Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In *Proc. of PKC 2003*, volume 2567 of *LNCS*, pages 31–46. Springer.

Boldyreva, A., Gentry, C., O'Neill, A., and Yum, D. (2007). Ordered multisignatures and identity-based sequential aggregate signatures, with applications to secure routing (extended abstract). In *Proc. of CCS 2007*, pages 276–285. ACM.

Boneh, D., Gentry, C., Lynn, B., and Shacham, H. (2003). Aggregate and verifiably encrypted signatures from bilinear maps. In *Proc. of EUROCRYPT 2003*, volume 2656 of LNCS, pages 416–432. Springer.

Broch, J., Maltz, D., Johnson, D., Hu, Y.-C., and Jetcheva, J. (1998). A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proc. of MobiCom 1998*, pages 85–97. ACM.

Gentry, C. and Ramzan, Z. (2006). Identity-based aggregate signatures. *Public Key Cryptography - PKC 2006*, 3958:257–273.

Ghosh, U. and Datta, R. (2011). Identity based secure aodv and tcp for mobile ad hoc networks. In *Proc. of ACWR 2011*, pages 339–346. ACM.

Ghosh, U. and Datta, R. (2013). Sdrp: Secure and dynamic routing protocol for mobile ad-hoc networks. *IET Network*, 3(3):235–243.

Guillemin, P. (2007). *ICTSB - RFID Networks Internet Of Things*. ETSI. http://docbox.etsi.org/Partners/ICTSB_Open/RFID/ICTSB_RFID_seminar_2007-10-24/P.Guillemin_ICTSB.

Hu, Y.-C., Perrig, A., and Johnson, D. (2002a). Ariadne: a secure on demand routing protocol for ad hoc network. In *Proc. of MobiCom 2002*. ACM.

Hu, Y.-C., Perrig, A., and Johnson, D. (2002b). Sead: Secure efficient distance vector routing for mobile wireless ad hoc networks. In *Proc. of WMCSA 2002*, pages 3–13. ACM.

Hu, Y.-C., Perrig, A., and Johnson, D. (2005). Ariadne: a secure on demand routing protocol for ad hoc network. *Wireless Networks*, 11:21–38.

Itakura, K. and Nakamura, K. (1983). A public-key cryptosystem suitable for digital multi-signatures. *NEC Research and Development*, 71:1–8.

Johnson, D., Hu, Y.-C., and Maltz, D. (2007). The dynamic source routing protocol (dsr) for mobile ad hoc networks for ipv4.

Johnson, D. and Maltz, D. (1996). Dynamic source routing in ad hoc wireless networks. *Mobile Computing*, 353:153–181.

Karlof, C. and Wagner, D. (2003). Secure routing in wireless sensor networks: Attacks and countermeasures. *Ad Hoc Networks*, 1(2-3):293–315.

Kim, J. and Tsudik, G. (2009). Srdp: Secure route discovery for dynamic source routing in manets. *Ad Hoc Networks*, 7(6):1097–1109.

Lysyanskaya, A., Micali, S., Reyzin, L., and Shacham, H. (2004). Sequential aggregate signatures from trapdoor permutations. In *Proc. of EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 74–90. Springer.

Micali, S., Ohta, K., and Reyzin, L. (2001). Accountable-subgroup multisignatures: extended abstract. In *Proc. of CCS 2001*, pages 245–254. ACM.

Muranaka, K., Yanai, N., Okamura, S., and Fujiwara, T. (2015). Secure routing protocols for sensor networks: Construction with signature schemes for multiple signers. In *Proc. of Trustcom 2015*, pages 1329–1336. IEEE.

Papadimitratos, P. and Haas, Z. J. (2002). Secure routing for mobile ad hoc networks. In *Proc. of CNDS 2002*, pages 27–31.

[1] https://www.nsnam.org/

Perkins, C. and Royer, E. (1999). Ad-hoc on-demand distance vector routing. In *Proc. of WMCSA 1999*, pages 90–100. IEEE.

Sanzgiri, K., LaFlamme, D., Dahill, B., Levine, B. N., Shields, C., and Belding-Royer, E. (2005). Authenticated routing for ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 23(3):598–610.

Shamir, A. (1984). Identity-based cryptosystems and signature schemes. In *Proc. of CRYPTO 84*, volume 196 of *LNCS*, pages 47–53. Springer.

Yanai, N., Manbo, M., and Okamoto, E. (2013). Ordered multisignature schemes under the cdh assumption without random oracles. In *Proc. of ISC 2013*, volume 7807 of LNCS, pages 367–377. Springer.

Zapata, M. and Asokan, N. (2002). Securing ad hoc routing protocols. In *Proc. of WISE*, pages 1–10. ACM Press.

Zhang, F., Jia, L., Basescu, C., Kim, T., Hu, Y., and Perrig, A. (2014). Mechanized network origin and path authenticity proofs. In *Proc. of ACM CCS 2014*, pages 346–357. ACM.

Zhou, L. and Haas, Z. (1999). Securing ad hoc network. *IEEE Network Magazine*, 13(6):24–30.