Regular Paper

Partially Doubly-Encrypted Identity-Based Encryption Constructed from a Certain Scheme for Content Centric Networking

Makoto Sato[1] Masami Mohri[2,a)] Hiroshi Doi[3,b)] Yoshiaki Shiraishi[4,c)]

Received: February 28, 2015, Accepted: September 2, 2015

Abstract: Information Centric Networking (ICN) is a promising paradigm for the future architecture of the Internet. Content Centric Networking (CCN) is an instantiation of the ICN paradigm. The challenging areas of CCN include congestion control, availability, security, etc. We focus on security, especially secure communications. Some schemes applying identity-based encryption (IBE) for content encryption over CCN have been proposed. However, such schemes generally have the key escrow problem that the private key generator which issues decryption keys to receivers can decrypt any ciphertext passively. We propose an IBE scheme approach to the problem by combining partial-double encryption, interest trace back, cut-through fragment forwarding and multi-path routing. Our scheme is IND-ID-CPA secure in the random oracle model.

Keywords: content-centric networking, secure communications, identity-based encryption, key escrow, semantic security

1. Introduction

The Information Centric Networking (ICN) paradigm [1] has attracted attention as the future architecture for the Internet. Content Centric Networking (CCN) is an instantiation of the ICN paradigm. In CCN, the connection between content and source is decoupled, and content becomes directly addressable by names, not by IP addresses. A user accesses data by sending a request packet (called Interest) containing the name of the content.

The challenging areas of CCN include congestion control, availability, security, etc. As for security, some schemes applying Identity-Based Encryption (IBE) have been proposed. In IBE, identities (ID) such as names, addresses, e-mail addresses or mobile phone numbers can be used as public keys, so there is no need for certificates as long as the validity of the ID is ensured. Usually, a public key encryption scheme is used to encrypt the session key, rather than to encrypt the contents. In this paper, to avoid confusion between a session key and a public/secret key, we use the notation M (message) for a session key.

Schemes in Refs. [2], [3] are hybrid schemes combining Hierarchical IBE (HIBE) [4] with Public Key Encryption (PKE). HIBE is used for content encryption and PKE for ensuring the validity of the system parameters generated by the Private Key Generator (PKG). However, IBE-based schemes including those in Refs. [2], [3] have the key escrow problem as one of the main problems. That is the PKG can decrypt any ciphertext passively because the private keys needed in decryption are all generated by this server. Kate and Goldberg [5] addresses this problem with an (n, t)-distributed PKG scheme so that multiple PKGs hold a share of the master secret key needed in generating private keys and each PKG cannot decrypt ciphertexts by itself. In order to solve the key escrow problem of IBE, we propose a devided ciphertext and partially doubly encrypted IBE scheme to use the distributed PKG. We make a comparison between our scheme and the (n, t)-distributed PKG schemes in Section 5.

The proposed scheme is an IBE scheme for CCN that approaches the key escrow problem with the following methods:

- Partial-double encryption,
- Interest trace back [6],
- Cut-through fragment forwarding [7], and
- Multi-path routing [8].

We ensure that the Service Providing Server (SPS) constructing our scheme and working as a cloud server cannot see the content by building an attack model and giving a security proof against the model. We also give a security overview for the case of the PKG attempting to decrypt ciphertexts. We organize the rest of the paper as follows. In Section 2, we describe basic components of the proposed scheme, then explain the scheme and define attack models in Section 3. We evaluate the security of the scheme in Section 4 before make a comparison in Section 5. Section 6 gives conclusions of this paper.

[1] Nagoya Institute of Technology, Aichi 466–8555, Japan
[2] Gifu University, Gifu 501–1193, Japan
[3] Institute of Information Security, Yokohama, Kanagawa 221–0835, Japan
[4] Kobe University, Kobe, Hyogo 657–8501, Japan
[a)] mmohri@gifu-u.ac.jp
[b)] doi@iisec.ac.jp
[c)] zenmei@port.kobe-u.ac.jp

2. Preliminaries

2.1 Content Centric Networking

In CCN, every node holds three tables: Forwarding Information Base (FIB), Pending Interest Table (PIT) and Content Store (CS). We just focus on FIB due to lack of space.

Consumer gets requested data as follows: Producer sends out announcements saying he has certain content. Then, according to the announcements, each router updates the FIB. The Consumer requests the data by publishing the *Interest*, which contains the name of the content. When the Interest arrives at a router, the router checks the FIB and forwards it to an appropriate direction. If a router has the requested content, the content is sent back in a *Data* packet through the reverse path of the Interest.

Content can be characterized as sharable or non-sharable based on its shareability property. With CCN's in-network caching mechanism, CCN provides efficiency of data dissemination for sharable content such as public web pages. On the other hand, each non-sharable content chunk should not be cached in CCN routers. A session key for establishing the encrypted channel is one of the non-sharable content.

2.2 Interest Trace Back

Dai et al. [6] have proposed Interest trace back as a counter measure against the Distributed Denial of Service (DDoS) attacks over CCN. The scheme traces back to the originator of the attacking Interest packets by sending back Data packets corresponding to the Interest.

2.3 Cut-through Fragment Forwarding

Content fragmentation seems unavoidable over CCN [7]. Cut-through fragment forwarding is a method that forwards individual content fragments without reassembly.

2.4 Multi-path Routing

CCN has support for multi-path routing [8]. For example, **Figure 1** shows that router R_1 and R_4 both have the same content named "*content/example*" and publish announcements (solid and dotted arrows), respectively. R_2 receives both announcements, and its FIB gets two entries. When R_2 publishes Interest for *content/example*, it will be forwarded through two paths: $R_2 \rightarrow R_1$ and $R_2 \rightarrow R_3 \rightarrow R_4$.

2.5 Bilinear Maps

Let $\mathbb{G}_1, \mathbb{G}_2$ be two cyclic groups of a prime order q. An admissible bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ satisfies the following properties for arbitrary $P, Q \in \mathbb{G}_1$, $a, b \in \mathbb{Z}_q^*$:

Bilinear: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$.

Non-degenerate: If $\hat{e}(P, Q) = 1$, $P = 0$ or $Q = 0$.

Computable: There are efficient algorithms to compute $\hat{e}(P, Q)$.

2.6 Boneh and Franklin's BasicIdent Scheme

Boneh and Franklin's identity-based encryption scheme [9] was proposed in 2001. There are two schemes. One has been proven to be IND-ID-CPA secure while the other has been proven to be IND-ID-CCA secure. These schemes are standardized in RFC5091 [10]. Our scheme is based on the former scheme, BasicIdent. BasicIdent is composed of the following four algorithms.

FIB of R_2

Name	Next
content/example	R_3
content/example	R_1

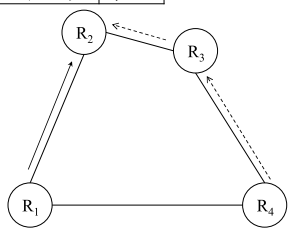

Fig. 1 Multipath support of CCN.

Setup: Given a security parameter $k \in \mathbb{Z}^+$, the algorithm works as follows:

Step 1: Output two groups $\mathbb{G}_1, \mathbb{G}_2$ of prime order q, and an admissible bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, and choose a random generator $P \in \mathbb{G}_1$.

Step 2: Pick a random $s \in \mathbb{Z}_q^*$ and compute $P_{pub} = sP$.

Step 3: Choose a cryptographic hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$, $H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^n$. $\mathbb{G}_1^*$ denotes the set $\mathbb{G}_1 \backslash \{O\}$, where O is the identity element in the group $\mathbb{G}_1$. The message space is $\mathcal{M} = \{0,1\}^n$ and the ciphertext space is $C = \mathbb{G}_1^* \times \{0,1\}^n$. The system parameters are $params = \langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, n, P, P_{pub}, H_1, H_2 \rangle$. The master key is $msk = s$.

Extract: For a given string $\mathrm{ID} \in \{0,1\}^*$ the algorithm does the following: 1) Compute $Q_{\mathrm{ID}} = H_1(\mathrm{ID}) \in \mathbb{G}_1^*$. 2) Set $d_{\mathrm{ID}} = sQ_{ID}$ as the private key corresponding to the ID.

Encrypt: To encrypt $M \in \{0,1\}^n$ with the identity $\mathrm{ID} \in \{0,1\}^*$, do the following: 1) Compute $Q_{\mathrm{ID}} = H_1(\mathrm{ID}) \in \mathbb{G}_1^*$. 2) Pick a random $r \in \mathbb{Z}_q^*$, and set the ciphertext to be $C = \langle rP, M \oplus H_2(\hat{e}(Q_{\mathrm{ID}}, P_{pub})^r) \rangle$

Decrypt: Let $C = \langle U, V \rangle \in C$ be a ciphertext encrypted with the identity ID. To decrypt C with the private key d_{ID} compute: $V \oplus H_2(\hat{e}(d_{\mathrm{ID}}, U)) = M$

We say that an identity-based encryption scheme is semantically secure (IND-ID-CPA) if no polynomially bounded adversary $\boldsymbol{A}$ has a non-negligible advantage against the Challenger in the following IND-ID-CPA game:

Setup: The challenger takes a security parameter k and runs the **Setup** algorithm. It gives the adversary the system parameters *params*. It keeps the master secret key to itself.

Phase 1: The adversary issues private key extraction queries $\mathrm{ID}_1, \cdots, \mathrm{ID}_m$. The challenger responds by running the algorithm **Extract** to generate the private key d_i corresponding to the identity ID_i. It sends d_i to the adversary. These queries could be asked adaptively.

Challenge: Once the adversary decides that **Phase 1** is over, it outputs two equal length messages M_0, $M_1 \in \mathcal{M}$ and an identity ID^* on which it wishes to be challenged. The only constraint is that ID^* did not appear in any private key extraction query in **Phase 1**. The challenger chooses a random bit $b \in \{0,1\}$ and sets $C = \mathbf{Encrypt}(params, \mathrm{ID}^*, M_b)$. It

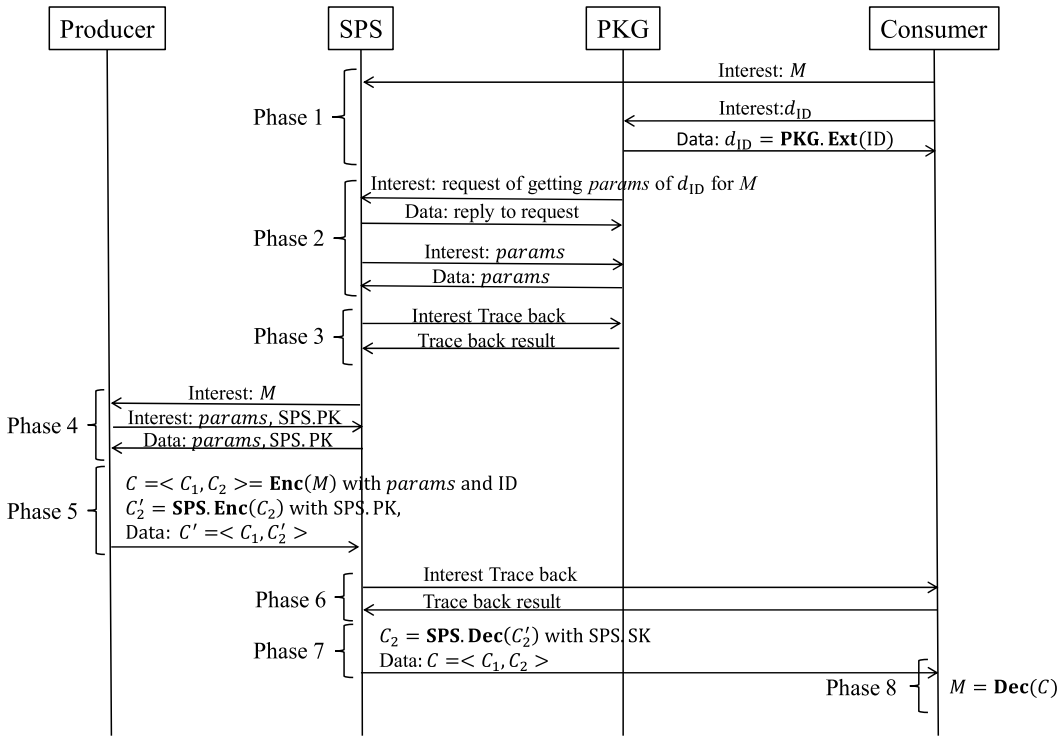

Fig. 2 Sequence diagram of the proposed scheme.

sends C as the challenge to the adversary.

Phase 2: The adversary issues more extraction queries $\mathrm{ID}_{m+1}, \cdots, \mathrm{ID}_n$. The only constraint is that $\mathrm{ID}_i \neq \mathrm{ID}^*$. The challenger responds as in **Phase 1**.

Guess: Finally, the adversary outputs a guess $b' \in \{0, 1\}$ and wins the game if $b = b'$.

3. Our Scheme

3.1 Entities

We define each entity constructing our scheme as follows.

Producer: To encrypt a message, uses the consumer's ID and generates the ciphertext $C = \langle C_1, C_2 \rangle$. Then generates the ciphertext $C' = \langle C_1, C_2' \rangle$ with the Service Providing Server's (SPS's) public key SPS.PK and sends C' to the SPS.

Consumer: To decrypt a ciphertext C, uses the private key d_{ID} sent from the PKG and generates the message M.

Service Providing Server (SPS): To decrypt C_2', uses SPS's private key SPS.SK and generates C_2. Then sends $C = \langle C_1, C_2 \rangle$ to the consumer.

Private key generator (PKG): To generate the private key d_{ID} corresponding to ID, uses the master secret key. Then sends the private key d_{ID} to the consumer over a secure channel.

CCN routers connect the entities and forward Interest and Data with multi-path routing and cut-through fragment forwarding aforementioned in Section 2.

3.2 Model

Our scheme works as follows. We show the sequence diagram of the scheme as **Fig. 2**. Receivers express their request for content using Interest packets, which are served for content discovery. Such Interest packets are routed based on the name of the requested content, using longest prefix matching. Suppose that each entity knows the prefix included in Interest to get data with others.

[Phase 1: Request]

Consumer requests the message M and the private key d_{ID} by sending the *Interest* packet to the SPS and the PKG. The PKG executes the following algorithms:

PKG.Setup: This algorithm takes as input a security parameter $k \in \mathbb{Z}^+$, and outputs the system parameters *params* and the master secret key *msk*.

PKG.Ext: This algorithm takes as input system parameters *params*, the master secret key *msk*, ID, and outputs the private key d_{ID}.

Then, the PKG replies d_{ID} as a *Data* packet to the Consumer.

[Phase 2: SPS Setup]

The PKG sends a request for the *params* of d_{ID} for M by sending *Interest* packet to the SPS. The SPS gets the system parameters *params* by sending an *Interest* and receiving *Data* to and from the PKG. Then, the SPS executes the following algorithm:

SPS.KG: This algorithm takes as input the system parameters *params*, and generates SPS's public key SPS.PK and secret key SPS.SK.

[Phase 3: Interest trace back 1]

The SPS requests the trace back to the PKG. Then the SPS conducts an interest trace back against the request from the PKG. The result will be used to check for disjoint paths at Phase 7.

[Phase 4: Producer setup]

The SPS sends a request for the *params* of d_{ID} and SPS.PK by sending an *Interest* packet to the Producer. The Producer gets the *params* and SPS.PK by sending *Interest* and receiving *Data* to and from the SPS.

[Phase 5: Message encryption]

The producer executes the following algorithms to encrypt a message M and to take steps against the key escrow problem by partially doubly encryption:

Enc: This algorithm, executed by a producer, takes as input the system parameters *params*, a message M and ID. It outputs the ciphertext $C = \langle C_1, C_2 \rangle$.

SPS.Enc: This algorithm, executed by a producer, takes C_2 and SPS's public key SPS.PK. It outputs the ciphertext C_2'.

Finally, the producer sends out $C' = \langle C_1, C_2' \rangle$ to the SPS as a *Data* packet corresponding to the *Interest* of M at Phase 4.

[Phase 6: Interest trace back 2]

The SPS executes a trace back against the *Interest* from the consumer. It checks out this result against the trace back result of Phase 3 and makes sure that the path between the SPS and the PKG is disjoint from the one between the SPS and the consumer. If so, the PKG is less likely to eavesdrop on the communication between the SPS and the consumer. Otherwise, the SPS stops this sequence and waits for a time out. Then it goes back to Phase 1.

[Phase 7: Partial decryption]

The SPS executes the following algorithm:

SPS.Dec: This algorithm takes as input C_2' and SPS's private key SPS.SK, and outputs C_2.

The SPS sends out C to the consumer in a multi-path routing and cut-through fragment forwarding fashion. Fragmented Cs are sent to the consumer through multipaths so that a third person is less likely to construct the whole C from fragmented Cs.

[Phase 8: Decryption]

The consumer executes the following algorithm:

Dec: This algorithm takes as input system parameters *params*, the private key d_{ID} corresponding to ID and a ciphertext C, to generate the message M.

3.3 Security Definitions

In order to make sure that the SPS cannot see content and the PKG is less likely to see content, we evaluate the following attack models:

- Attack model 1: The PKG tries to decrypt ciphertexts.

We just show a security overview against this attack model.

- Attack model 2: The SPS colludes with an adversary

We prove that the proposed scheme satisfies IND-ID-CPA security against the attack model.

3.3.1 Attack Model 1 and Security in the Model

Security in this model is defined as the following game:

Setup: The challenger runs **SPS.KG** and **PKG.Setup**. Then the challenger gives an adversary SPS.PK, *params* and *msk*.

Challenge: The adversary outputs two equal length message M_0, M_1, and an identity ID^* on which it wishes to be challenged. The challenger chooses a random $\sigma \in \{0,1\}$ and sets $C^* = \mathbf{Enc}(M_\sigma, \mathrm{ID}^*)$. It sends C^* as the challenge to the adversary.

Guess: Finally, the adversary outputs a guess $\sigma' \in \{0,1\}$ and wins the game if $\sigma = \sigma'$. If the probability that the adversary wins the game is $1/2 + \varepsilon(k)$, the adversary has the advantage $\varepsilon(k)$.

Definition 1 We say that our scheme is IND-CPA secure, if no polynomial time adversary has a non-negligible advantage $\varepsilon(k)$ against the challenger.

3.3.2 Attack Model 2 and Security in the Model

In this model, we assume that the SPS colludes with an adversary and that cloud servers are "honest-but-curious" [11]. This means cloud servers will follow our proposed protocol in general, but try to find out as much secret information as possible based on their inputs. By proving the security of the proposed scheme based on this assumption, we show that the SPS can be used as cloud servers. Security in this model is defined with the following game:

Setup: The challenger runs **SPS.KG** and **PKG.Setup**, then the challenger gives an adversary SPS.PK, SPS.SK and *params*.

Phase 1: The adversary can issue the following query:

Extraction query: The adversary sends ID to the challenger, then the challenger responds by running the algorithm **PKG.Ext** to generate the private key d_{ID} corresponding to the identity ID. It sends d_{ID} to the adversary.

Challenge: Once the adversary decides that **Phase 1** is over, it outputs two equal length messages M_0, M_1 and an identity ID^* on which it wishes to be challenged. There is a constraint that ID^* did not appear in any extraction query in **Phase 1**. The challenger chooses a random $\sigma \in \{0,1\}$ and sets $C^* = \langle C_1^*, C_2^* \rangle = \mathbf{Enc}(params, M_\sigma, \mathrm{ID}^*)$. It sends C^* as the challenge to the adversary.

Phase 2: The adversary can issue the same query in **Phase 1**. The only constraint is that $\mathrm{ID} \neq \mathrm{ID}^*$.

Guess: Finally, the adversary outputs a guess $\sigma' \in \{0,1\}$ and wins the game if $\sigma = \sigma'$. If the probability that the adversary wins the game is $1/2 + \varepsilon(k)$, the adversary has the advantage $\varepsilon(k)$.

Definition 2 We say that our scheme is IND-ID-CPA secure, if no polynomial time adversary has a non-negligible advantage $\varepsilon(k)$ against the challenger.

3.4 Concrete Construction

We show a concrete construction of our scheme as follows:

PKG.Setup: Given a security parameter $k \in \mathbb{Z}^+$, the algorithm works as follows:

Step 1: Output two groups $\mathbb{G}_1$, $\mathbb{G}_2$ of prime order q, and an admissible bilinear map $\hat{e}: \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$, and choose a random generator $P \in \mathbb{G}_1$.

Step 2: Pick a random $s \in \mathbb{Z}_q^*$ and compute $P_{pub} = sP$.

Step 3: Choose cryptographic hash functions $H_1 : \{0,1\}^* \to \mathbb{G}_1^*$, $H_2 : \mathbb{G}_2 \to \{0,1\}^n$, $H_M : \mathbb{G}_1 \to \{0,1\}^n$. The message space is $\mathcal{M} = \{0,1\}^n$ and the ciphertext space is $C = \mathbb{G}_1^* \times \{0,1\}^n$. The system parameters are $params = \langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, n, P, P_{pub}, H_1, H_2, H_M \rangle$. The master-key is $msk = s$.

PKG.Extract: Given a string $\mathrm{ID} \in \{0,1\}^*$ and the master key s, the algorithm works as follows:

Step 1: Compute $Q_{\mathrm{ID}} = H_1(\mathrm{ID}) \in \mathbb{G}_1^*$.

Step 2: Set $d_{\mathrm{ID}} = sQ_{\mathrm{ID}}$ as the private key corresponding to the ID.

SPS.KG: Given $b \in \mathbb{Z}_q^*$, the algorithm sets $(\mathrm{SPS.PK}, \mathrm{SPS.SK}) =$

(bP, b). SPS.PK and SPS.SK are SPS's public key and SPS's secret key, respectively.

Enc: To encrypt $M \in \mathcal{M}$ with the identity ID, do the following:
Step 1: Compute $Q_{\mathrm{ID}} = H_1(\mathrm{ID}) \in \mathbb{G}_1^*$.
Step 2: Pick a random $r \in \mathbb{Z}_q^*$, and set the ciphertext to be $C = \langle C_1, C_2 \rangle = \langle rP, M \oplus H_2(\hat{e}(Q_{\mathrm{ID}}, P_{pub})^r) \rangle$.

SPS.Enc: Given C_2, the algorithm works as follows:
Step 1: Pick a random $a \in \mathbb{Z}_q^*$, and set $C'_{M1} = aP$.
Step 2: Set $C'_{M2} = C_2 \oplus H_M(abP)$.
Step 3: Set $C'_2 = \langle C'_{M1}, C'_{M2} \rangle$.

SPS.Dec: Given a message component C'_2 and SPS.SK $= b$, sets $C_2 = C'_{M2} \oplus H_M(bC'_{M1})$.

Dec: To decrypt C with the private key d_{ID}, computes: $M = C_2 \oplus H_2(\hat{e}(d_{\mathrm{ID}}, C_1))$.

4. Security

We show a security overview against the attack model 1 and prove the security of the proposed scheme in attack model 2, as defined in Subsection 3.4.

4.1 Security against Attack Model 1

The PKG, of course, can generate any private key in this model, so the security of the scheme depends on how many fragments of C it collects.

Suppose that $C = \langle C_1, C_2 \rangle$ is N bit ($C_1 = N_1$ bit, $C_2 = N_2$ bit) and fragmented into n parts $\langle c_1, c_2, \ldots, c_n \rangle$, where each part c_i is N/n bit. When the PKG corrects l parts, there can be $2^{(n-l)N/n}$ kinds of C from the viewpoint of the PKG. We assume that C_1 is basically, uniformly fragmented into n parts. Although the size of n is specified by CCN routers, producers can control the size of N by choosing a large message space and an appropriate hash function $H_2(.)$, where $C_2 = M \oplus H_2(.)$. Therefore, we believe that the presumption of the whole C is infeasible in polynomial time as long as the size of N is large enough.

The worst case is that a certain c_i contains the whole C_1 (for example, $c_1 = C_1$, $c_2 = [C_2]_1^{N/n}$, $c_3 = [C_2]_{(N/n)+1}^{2N/n}$, …, $c_n = [C_2]_{N_2-(N/n)+1}^{N_2}$, where $[X]_j^k$ means from the j th to the k th bit of X) and the PKG gets it. In most IBE schemes, the PKG is capable of computing an input for $H_2(.)$ from C_1. In that case, PKG can obtain partial messages by computing $[M]_{(i-2)(N/n)+1}^{(i-1)(N/n)} = [C_2]_{(i-2)(N/n)+1}^{(i-1)(N/n)} \oplus [H_2(.)]_{(i-2)(N/n)+1}^{(i-1)(N/n)} (i = 2, 3, \ldots)$ corresponding to any fragment $c_i = [C_2]_{(i-2)(N/n)+1}^{(i-1)(N/n)}$ it obtains.

This is still an open problem, but there is the possibility of resolving this concern by applying an all-or-nothing transform method [12].

The result implies that if a combination of techniques of interest trace back, multi-path routing and cut-through fragment forwarding work together, the proposed scheme is secure against attack model 1.

4.2 Security Proof against Attack Model 2

Theorem 1 Suppose Boneh and Franklin's BasicIdent scheme is IND-ID-CPA secure, then our scheme is also IND-ID-CPA secure against attack model 2 in the random oracle model.

Proof Let $\boldsymbol{A}$ be an IND-ID-CPA adversary with advantage $\varepsilon(k)$ against our scheme. We prove that there is an adversary $\boldsymbol{B}$ which has an advantage at least $\varepsilon(k)$ against a BasicIdent scheme simulator (Given the input, it responds according to algorithms of the scheme). $\boldsymbol{B}$ works by interacting with $\boldsymbol{A}$ in the IND-ID-CPA game as follows:

Setup: The adversary $\boldsymbol{B}$ receives *params* from the simulator, then runs **SPS.KG**. It gives the adversary *params*, SPS.PK, and SPS.SK to the adversary $\boldsymbol{A}$.

Phase 1: The adversary $\boldsymbol{A}$ issues extraction queries. The adversary $\boldsymbol{B}$ receives the private key d_{ID} corresponding to the identity ID from the simulator, then it gives d_{ID} to the adversary $\boldsymbol{A}$.

Challenge: The adversary $\boldsymbol{A}$ sends $\langle \mathrm{ID}^*, M_0, M_1 \rangle$ to the adversary $\boldsymbol{B}$, then $\boldsymbol{B}$ sends the simulator them. The simulator picks a random $\sigma \in \{0, 1\}$, and sets $C^* = \langle C_1^*, C_2^* \rangle =$ **Encrypt**$(params, M_\sigma, \mathrm{ID}^*)$. It sends C^* to adversary $\boldsymbol{B}$. Finally, adversary $\boldsymbol{B}$ sends C^* to adversary $\boldsymbol{A}$.

Phase 2: The adversary $\boldsymbol{A}$ issues the same extraction queries in **Phase 1**. The only constraint is that ID $\neq$ ID*.

Guess: Finally, the adversary $\boldsymbol{A}$ outputs $\sigma' \in \{0, 1\}$. The adversary $\boldsymbol{B}$ outputs σ' as a guess.

The simulation above shows there is an adversary $\boldsymbol{B}$ that has an advantage at least $\varepsilon(k)$ against the BasicIdent scheme simulator if there is an adversary $\boldsymbol{A}$ that has an advantage $\varepsilon(k)$ against our scheme.

Owing to doubly encrypting, the PKG cannot get a session key. Because the SPS can only partially decrypt a ciphertext (i.e., the SPS can get an IBE ciphertext), SPS also cannot get M.

5. Comparison

We compare our scheme with (n,t)-distributed PKG schemes [5]. In (n, t)-distributed PKG schemes, a user has to contact at least t + 1 PKGs and get the private key shares in order to generate the private key. Every channel between a user and the PKGs has to be a secure one.

One of basic security concepts of CCN is built-in security [3] which means the content itself has a mechanism for content protection. Therefore, it is desirable that the necessity for secure channels is as little as possible. As to this point, the proposed scheme needs just one secure channel. However, our scheme does not address the single point of failure problem while (n, t)-distributed PKG schemes do.

In our model, in addition to the traditional IBE, the SPS is only added. We evaluate the increased computation cost related to the SPS. The sender's increasing cost is related to "encrypting" the IBE ciphertext. This is evaluated as one PKE encryption cost. With regard to the SPS, (1) one decrypting cost, and (2) one communication cost about sending the partial decryption result to a consumer with not an encrypted communication channel (e.g., SSL/TLS) but multi-path routing and cut-through fragment forwarding increase. The decrypting cost would not significantly affect, because it is almost same as the computation cost of SSL/TLS. If the multi-path routing and cut-through fragment forwarding are in basic functions, the cost would be negligible.

6. Conclusion

CCN has challenging areas including congestion control, availability, security, etc. We focused on security, especially secure communications. Some schemes combining HIBE and PKE have been proposed as ways of content protection over CCN. Such schemes use HIBE for content encryption and PKE for ensuring the validity of the system parameters generated by PKG. However, those schemes have the key escrow problem – the PKG can decrypt any ciphertext passively because every private keys needed in decryption are generated by this server.

In this paper, we proposed an IBE approach to the key escrow problem by combining partial-double encryption, interest trace back, cut-through fragment forwarding, and multi-path routing. Partial-double encryption enables the content producer to be offline before all transactions are over, and interest trace back is used for making sure that the PKG is less likely to eavesdrop the paths through which content passes. Cut-through fragment forwarding and multi-path routing prevent a third person from assembling the whole content. We evaluated the security of the proposed scheme by defining some attack models and giving security proof against the models. We showed that our scheme needs less secure channels than (n,t)-distributed PKG schemes.

Although the proposed scheme is based on a pair of Interest and Data packets so as to work on CCN architecture and is IND-ID-CPA secure in the random oracle model theoretically, we have not confirmed the actual behavior of the proposed scheme over networks. Further studies are needed in order to evaluate the performance of the proposed scheme by considering interest trace back, multi-path routing and cut-through fragment forwarding.

Acknowledgments This work was supported by JSPS KAKENHI Grant Number 25330151.

References

[1] Jacobson, V., Smetters, D.K., Thornton, J.D., et al.: Networking Named Content, *Proc. 5th International Conference on Emerging Networking Experiments and Technologies*, ACM (2009).
[2] Hamdane, B., Serhrouchni, A., Fadlallah, A. and Fatmi, S.G.E.: Named-Data Security Scheme for Named Data Networking, *Network of the Future* (*NOF*), pp.1–6 (2012).
[3] Zhang, X., Chang, K., Xiong, H., et al.: Towards Name-based Trust and Security for Content-centric Network, *Network Protocols* (*ICNP*), pp.1–6 (2011).
[4] Gentry, C. and Silverberg, A.: Hierarchical ID-Based Cryptography, *Advances in Cryptology CRYPTO 2002*, pp.149–155 (2002).
[5] Kate, A. and Goldberg, I.: Distributed Private-Key Generators for Identity-Based Cryptography, *Proc. 7th International Conf. Security and Cryptography for Networks*, Amalfi, Italy, pp.436–453 (2010).
[6] Dai, H., Wang, Y., Fan, J. and Liu, B.: Mitigate DDoS Attacks in NDN by Interest Traceback, *Proc. IEEE INFOCOM NOMEN Workshop*, pp.381–386, IEEE Press (2013).
[7] Ghali, C., Narayanan, A., Oran, D. and Tsudik, G.: Secure Fragmentation for Content-Centric Networks, arXiv:1405.2861 (2014).
[8] Dai, H., Lu, J., Wang, Y. and Liu, B.: A Two-layer Intra-domain Routing Scheme for Named Data Networking, *Global Communications Conference* (*GLOBECOM*), pp.2815–2820 (2012).
[9] Boneh, D. and Franklin, M.: Identity-Based Encryption from the Weil Pairing, *SIAM Journal of Computing*, Vol.32, No.3, pp.586–615 (2003).
[10] Network Working Group.: Identity-Based Cryptography Standard (IBCS) #1: Supersingular Curve Implementations of the BF and BB1 Cryptosystems, available from ⟨http://tools.ietf.org/html/rfc5091⟩.
[11] Vimercati, S.D.C.D, Foresti, S., Jajodia, S., Paraboschi, S. and Samarati, P.: Over-encryption: Management of Access Control Evolution on Outsourced Data, *Proc. 33rd International Conference on Very Large Data Bases* (*VLDB'07*), pp.123–134 (2007).
[12] Boyko, V.: On the Security Properties of OAEP as an All-or-Nothing Transform, *Advances in Cryptology Crypto 99*, pp.503–518 (1999).

Makoto Sato received B.E. degree from Nagoya Institute of Technology, Japan, in 2013. He is a graduate student of the same institute. His current research interests include information security and cryptography. He is a student member of IPSJ.

Masami Mohri received B.E. and M.E. degrees from Ehime University, Japan, in 1993 and 1995 respectively. She received a Ph.D. degree in Engineering from the University of Tokushima, Japan in 2002. From 1995 to 1998 she was an assistant professor at the Department of Management and Information Science, Kagawa Junior College, Japan. From 1998 to 2002 she was a research associate of the Department of Information Science and Intelligent Systems, University of Tokushima, Japan. From 2003 to 2008 she was a lecturer of the same department. Since 2008, she has been an associate professor at the Information and Multimedia Center, Gifu University, Japan. Her research interests are in coding theory, information security and cryptography. She is a member of IEEE and a senior member of IEICE.

Hiroshi Doi received B.S. degree in mathematics from Okayama University in 1988, M.S. degree in information science from JAIST in 1994, and D.S. degree from Okayama University in 2000, respectively. He is currently a professor at the Graduate School of Information Security, Institute of Information Security, Japan. His research interests include information security and cryptography.

Yoshiaki Shiraishi received B.E. and M.E. degrees from Ehime University, Japan, and Ph.D. degree from University of Tokushima, Japan, in 1995, 1997 and 2000, respectively. From 2002 to 2006 he was a lecturer at the Department of Informatics, Kinki University, Japan. From 2006 to 2013 he was an associate professor at the Department of Computer Science and Engineering, Nagoya Institute of Technology, Japan. Since 2013, he has been an associate professor at the Department of Electrical and Electronic Engineering, Kobe University, Japan. His current research interests include information security, cryptography, computer network, and knowledge sharing and creation support. He received the SCIS 20th Aniversary Award and the SCIS Paper Award from the ISEC group of IEICE in 2003 and 2006, respectively. He received the SIG-ITS Excellent Paper Award from SIG-ITS of IPSJ in 2015. He is a member of IEEE, ACM and a senior member of IEICE.