# Identity-Based Broadcast Encryption with Outsourced Partial Decryption for Hybrid Security Models in Edge Computing

Jongkil Kim
University of Wollongong
Wollongong, Australia
jongkil@uow.edu.au

Seyit Camtepe
Data61/CSIRO
Marsfield, Australia
Seyit.Camtepe@data61.csiro.au

Willy Susilo
University of Wollongong
Wollongong, Australia
wsusilo@uow.edu.au

Surya Nepal
Data61/CSIRO
Marsfield, Australia
Surya.Nepal@data61.csiro.au

Joonsang Baek
University of Wollongong
Wollongong, Australia
baek@uow.edu.au

## ABSTRACT

Each layer of nodes and communication networks in edge computing, from cloud to the end device (i.e, often considered as resource-constrained IoT devices), exhibits a different level of trust for each stakeholder - e.g., edge nodes may not be fully trusted by IoT devices and the cloud. Moreover, asymmetric nature of resources between layers makes it hard to establish a balance between security and performance - e.g., lightweight cryptography may degrade security level against untrusted nodes while heavyweight ones may not be feasible for the light-weight end devices. An advanced encryption scheme such as the Identity-Based Broadcast Encryption (IBBE) is a popular technique to reduce storage and communication overhead. However, IBBE requires heavy computation to the end devices and still does not fully satisfy the security requirements that exist in the layers of edge computing. This paper presents a new IBBE with outsourced partial decryption for hybrid security models that each layer in edge computing requires. It balances the computational overhead based on asymmetric nature that nodes in each layer have. Particularly, with new schemes, the ciphertext can be transformed from its initial format. The cloud encrypts their data for multiple end devices and store them in the edge nodes, but those interim nodes can blindly transform the ciphertext from the cloud into a form which (i) is decryptable by only an authorized end device, and (ii) imposes smaller decryption and data transmission burden to end devices, regardless of the number of recipients. Our security analysis shows that new schemes are selectively and adaptively secure. We implement our solution and show that new schemes reduce the communication overhead from an edge node to end devices and the computation overhead on the end devices, compared to the original IBBE schemes.


*AsiaCCS '19, July 9–12, 2019, Auckland, New Zealand*

ACM ISBN 978-1-4503-6752-3/19/07…$15.00
https://doi.org/10.1145/3321705.3329825

## CCS CONCEPTS

• **Security and privacy → Public key encryption**; *Security protocols.*

## KEYWORDS

Identity-based encryption, Identity-based broadcast encryption, Outsourced decryption, Edge computing

**ACM Reference Format:**
Jongkil Kim, Seyit Camtepe, Willy Susilo, Surya Nepal, and Joonsang Baek. 2019. Identity-Based Broadcast Encryption with Outsourced Partial Decryption for Hybrid Security Models in Edge Computing. In *ACM Asia Conference on Computer and Communications Security (AsiaCCS '19), July 9–12, 2019, Auckland, New Zealand.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3321705.3329825

## 1 INTRODUCTION

Edge computing [3] is a modern paradigm to provide a cloud-like service at the edge of the network. Unlike the typical cloud-only framework, in the edge computing, data are located near the end users so that it reduces the unnecessary round-trips of the data transmission and provides a better quality of service by serving the users based on their locational proximity [2, 26]. Edge computing has a layered service architecture (see Figure 1) which efficiently supports large scaled IoT networks consisting of sensors, actuators, mobiles and connected vehicles. A cloud server and an edge node exist in different layers and can be provided by different vendors.

Particularly, the cloud service providers usually own a few large centralized data centers to handle a large amount of data and offer various services. On the other hand, the "edges" are located near the network edges, where IoT devices are connected (almost) directly. The edges are likely owned by local companies such as regional telecommunication companies (so-called "telcos"), which own a number of base stations or public access points in the regional area, which can be used as edge servers. Due to this difference in the nature of service they provide, telecommunication companies and cloud service providers usually work under a service level agreement (SLA) to cooperate.

In spite of the SLA, edges cannot be fully trusted. The edge service providers (telcos) may be curious about the users' data and want to use them for their own purposes such as marketing or advertisement. More serious problem is that the edges are located

in physically vulnerable places, to which malicious users could get access. Unlike the databases managed by the cloud service providers, the data storage of edges can be taken physically or directly accessed by an adversary due to their proximity to end users.

Data privacy is one of the major concerns for cloud service providers when they outsource the data to the other servers being operated independently [23, 24]. A cloud service provider needs to protect users' data stored in the edge. One of the most promising protection methods is the data encryption - the cloud creates and distributes decryption keys to the end users of the system to ensure that the data is hidden from the edge. However, applying encryption to the data stored in the cloud or edge servers introduces an overhead.

Compared to the cloud, the edge has much smaller data storage and the data stored in the edge can be localized information shared by multiple users. Public key encryption is often considered to encrypt data for multiple users due to its simple key management mechanism and efficiency. Identity-based broadcast encryption (IBBE) [4, 21] and attribute-based encryption [8, 20] are well-known public key encryption systems allowing multiple users to efficiently share the secret data. Those primitives are rooted in identity-based encryption (IBE) [22], where each user has their own key associated with its identity and ciphertexts can be decrypted by only an entitled recipient.

Although efficiency of public key encryption schemes is rapidly improving, they are still too heavy to be deployed in edge computing. They require intensive computation such as exponentiations or scalar multiplications, depending on the group structure, or pairing computation. Moreover, the number of those operations for decryption usually increases as the number of recipients grows. In edge computing, however, end devices are usually resource-constrained. They are often battery-powered and have a slow processor. Building a light-weight broadcast-encryption scheme which can share secret data without demanding computational burden on the end device and handle multi-recipients efficiently (for example, providing constant-sized information for the decryption) remains an important yet challenging task in the edge computing framework.

## 1.1 Our Contribution

We propose a new cryptographic system, IBBE with outsourced partial decryption (IBBE-OPD) for edge computing. Our new system enables edge computing to use an identity-based broadcast encryption for resource-constrained end devices such as IoT devices. Particularly, it minimizes the resources required to decrypt ciphertexts and improve the security between an edge and an end device. To achieve this, we let IBBE-OPD have two types of ciphertexts. One type of ciphertext is stored in the edge and enables end devices to share encrypted data. This ciphertext makes the data hidden from the edge and others who are not entitled to see the data. The other type of ciphertext is transformed from the original ciphertext on the fly by the edge. It is transmitted from the edge to the end devices. This ciphertext minimizes the resources used for decryption and makes our suggested schemes suitable for the edge computing.

Particularly, our IBBE-OPD enhances edge's access control capability without any extra authentication and one-to-one encryption. In the existing outsourced decryption schemes [9, 15], all recipients share a common key to decrypt a ciphertext partially decrypted by a third party decryption oracle (i.e. an edge or proxy server). Since it is a shared common key, any end device who possesses a private key can decrypt the partially decrypted ciphertext regardless of their identity or attributes. This implies that the decryption oracle needs to authenticate an end device before it partially decrypts data for the device. Moreover, it can only send the partially decrypted data through an one-to-one encrypted channel to prevent other end devices' unauthorized decryption.

However, in our systems, the partially decrypted ciphertext sent by an edge to an end user is a ciphertext of IBE which only can be decrypted by the key associated with the recipient's identity. Therefore, the edge can control the actual recipient of the encrypted data even in the open broadcast network, without any extra encrypted channel. Also, the possession of the private key proves the identity of the end device. It means that the edge does not require additional authentication to end devices for access control.

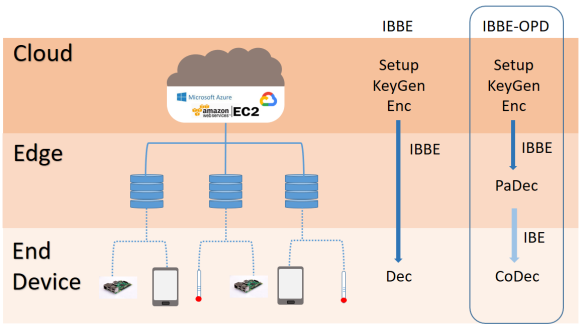
In addition to the security enhancement, compared to the IBBE schemes, our new schemes bring the following advantages in the edge computing framework:

- Our scheme successfully reduces duplication in the edge. The size of the ciphertext of our system is constant and small. It does not grow as the number of recipients increases. Therefore, the encryption overhead in the edge's data storage is independent from the number of recipients.
- The transmission between an edge and an end device is minimal. Unlike general broadcast encryption schemes, our scheme requires only a few constant-sized group elements for end users without any additional parameter which may grow linearly to the number of recipients.
- The computation burden for the decryption in an end device is constant and small, i.e. the only computationally heavy operation required to the end user is a single pairing computation.
- Our scheme requires very small sized secure memory to store the private key. The size of a private key is only a single group element (and an additional bit only for the adaptive scheme).

Based on the aforementioned improvements, the system entities such as the cloud, edges and end devices have the following advantages in our suggested scheme:

- **Cloud**: The cloud can keep the data secret and minimize the data usages in the edge. Also, it can control the list of the data recipients by collaborating with the edge server with less key management effort.
- **Edge**: The edge can reduce the data storage overhead caused by the encryption. The multiple users share the same ciphertext which is constant and small. The edge can control the access of data recipients and trace the recipients without any extra authentication or one-to-one encryption such as TLS, WPA or WPA2.
- **End device**: The end device can enjoy the efficiency. The transmission and computation required for end devices are constant and small. This may prolong the battery life and reduce the latency caused by decryption. Moreover, the end devices keep only

**Figure 1: IBBE-OPD on Edge Computing**




a constant-sized secret key in secure memory without needing complex key management procedure.

We achieve such efficient schemes by dividing a decryption process into two: When an end device downloads the data, the edge that keeps the encrypted content (i.e. IBBE ciphertext) will perform partial decryption. As the result of the partial decryption, the edge transforms the stored IBBE ciphertext for multiple receivers to the ciphertext of an IBE scheme, in which all receivers except the device requesting the data were revoked. Unlike the general revocation technique, our outsourced partial decryption transforms a ciphertext to a form imposing the less computational burden to the recipient due to decryption.

Particularly, we let Setup (**Setup**), Key Generation (**KeyGen**), Encryption (**Enc**) and Decryption (**Dec**) be the algorithms that consist of the traditional IBBE. We divide the decryption algorithm of the traditional broadcast encryption system, **Dec**, into two algorithms, which are the partial decryption (**PaDec**) and complete decryption (**CoDec**). In edge computing system, **PaDec** is computed by edges. **PaDec** does not require knowledge of the end device's private key, but knows its identity. **CoDec** is the decryption process which must be carried out by the end device possessing the private key associated with its unique identity. We depict the layers of edge computing and IBBE-OPD operations in Figure 1.

As instances of our IBBE-OPD, we provide two schemes, one with semi-static security and another with adaptive security. Both schemes are tailored for edge computing using outsourced partial decryption technique.

## 2 RELATED WORK

The concept of broadcast encryption was introduced by Fiat and Naor [6]. In broadcast encryption [5, 6, 18], a ciphertext can be shared and decrypted by multiple receivers. An efficient broadcast encryption scheme which has a short ciphertext and a private key with fully collusion resistance was firstly introduced by Boneh et al. [1]. Identity-based broadcast encryption was introduced independently by Delerablée [4] and Sakai and Furukawa [21]. The first adaptively secure IBBE scheme was introduced by Gentry and Waters [7]. First, they introduced a semi-static secure scheme which has constant sized ciphertexts and private keys. They, then, showed the conversion technique from semi-static security to adaptive security using two key technique introduced by Katz and Wang [11]. More recently, Jongkil et al. [12] improved the efficiency of Gentry and Waters' IBBE scheme by removing tags and simplifying the security proof. IBBE schemes were evolved to support more advanced functionality such as anonymity [14, 16, 25] and more complex functions such as Attribute-Based Encryption [8, 20].

The outsourced decryption [9, 10, 13, 15, 19, 27] improves the efficiency of an overall system by using a third party decryption oracles without revealing the secret. After a seminal introduction from Green at al. [9], outsourced decryption is actively researched for public key encryption, particularly, for attribute-based encryption. In those outsourced decryption schemes, all end devices share a common decryption key which can decrypt a partially decrypted ciphertext from a decryption oracle. A disadvantage of this approach is that the common shared key is known to all end devices in the system, even to the one who are not entitled to decrypt ciphertext. Due to this, the partially decrypted data must be computed after the proxy confirmed the end device's attributes or identity through an authentication mechanism. Moreover, it must be transmitted from the proxy server to the end device through a secure channel due to the same reason. Verifiable outsourced decryption schemes [10, 13, 15, 19] add an extra verification feature to the outsourced decryption which enables the end device to ensure the correctness of the partial decryption, but these schemes still have the aforementioned disadvantages.

All those schemes are based on public key cryptography. Symmetric key cryptography also used to share data contents among multiple parties (e.g, receivers such as users or devices).

## 3 PRELIMINARIES

### 3.1 Bilinear Pairing

Let set $\mathcal{G}$ as a group generator taking a security parameter $\lambda$ as input and outputting a description of a bilinear group $\mathcal{G}$. For our purposes, we will have $\mathcal{G}$ output $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ where $p$ are a prime, $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are cyclic groups of order $p$, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is an efficiently computable non-degenerate bilinear map. We assume that the group operations in $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ as well as the bilinear map $e$ are efficiently computable in polynomial time with respect to $\lambda$ and that the group descriptions of $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ include generators of the respective cyclic groups. If $\mathbb{G}_1 = \mathbb{G}_2$, we call $e$ a symmetric pairing and we use $\mathbb{G}$ to denote both $\mathbb{G}_1$ and $\mathbb{G}_2$ (i.e. $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$). Otherwise, we call $e$ an asymmetric pairing.

### 3.2 Assumption

**Definition 1.** *The Decision Bilinear Diffie-Hellman Exponent Sum (DB-DHES) Problem for (S, m) [7]. Fix $Z \subset \mathbb{Z}$ and $m \in \mathbb{Z} \setminus Z$. Let $\mathbb{G}$ and $\mathbb{G}_T$ be groups of order $p$ with a bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$, and let $g$ be generators for $\mathbb{G}$. Set $\alpha \xleftarrow{R} \mathbb{Z}_p^*$ and $b \xleftarrow{R} \{0, 1\}$. If $b = 0$, set $T \leftarrow e(g, g)^{\alpha^m}$; otherwise, set $T \xleftarrow{R} \mathbb{G}_T$. Ouput*

$$\{g^{\alpha^i} : i \in Z\} \text{ and } T.$$

*The problem is to guess $b$.*

We define the advantage of the adversary of $\mathcal{A}$ to guess $b$ correctly as follows:

$$Adv_{\mathcal{A},m}^{DHES}(\lambda) = \left| \Pr[b = b'] - \frac{1}{2} \right|.$$

### 3.3 IBBE with Outsourced Partial Decryption

Our IBBE-OPD consists of five algorithms, setup (**Setup**), private key generation (**KeyGen**), encryption (**Enc**), partial decryption (**PaDec**)and complete decryption (**CoDec**) as defined below :

- **Setup**($\lambda$, $n$, $\ell$) takes as input the number of recipients $n$ and the maximum size of a broadcast recipient group $\ell$ ($\leq n$). It outputs a public/master secret key pair $\langle PK, MSK\rangle$.
- **KeyGen**($i$, $MSK$) takes as input an index $i \in \{1, ..., n\}$ and the secret key $MSK$. It outputs a private key $sk_i$.
- **Enc**($S$, $PK$) takes as input a subset $S \subseteq \{1, ..., n\}$ and a public key $PK$. If $|S| \leq \ell$, it outputs a pair $\langle Hdr, K\rangle$ where $Hdr$ is called the header and $K \in \mathcal{K}$ is a message encryption key.
- **PaDec**($S$, $i$, $Hdr$, $PK$) takes as input a subset $S \subseteq \{1, ..., n\}$, an identity $i$, a header $Hdr$, and the public key $PK$. If $|S| \leq \ell$ and $i \in S$, then the algorithm outputs the partially decrypted header $PaHdr$.
- **CoDec**($sk_i$, $PaHdr$, $PK$) takes as input a private key $sk_i$ for $i$, a partially decrypted header $PaHdr$, and the public key $PK$. If $PaHdr$ is a partial decrypted header for $i$, then the algorithm outputs the message encryption key $K \in \mathcal{K}$.

**Correctness Property.** For the correctness, the following property must be satisfied.

Let $(PK, MSK) \leftarrow$ **Setup**$(\lambda, n, \ell)$ and $sk_i \leftarrow$ **KeyGen**$(i, MSK)$ for a user $i$. For a set of identities $S$ where $|S| \leq \ell \leq n$, and $(Hdr, K) \leftarrow$ **Enc**$(S, PK)$. If $i \in S$, $PaHdr \leftarrow$ **PaDec**$(S, i, Hdr, PK)$. Then, $K \leftarrow$ **CoDec**$(sk_i, PaHdr, PK)$.

### 3.4 Security Models

Our new schemes have two types of the ciphertexts. One is the output of the encryption algorithm **Enc** and can be decrypted by multiple parties, the same as the original IBBE. The other one is the output of the partial decryption algorithm **PaDec** where the ciphertext can be decrypted only by a single party which possesses a valid private key, the same as the typical IBE scheme. Therefore, our schemes must satisfy the security models of both IBBE and IBE at the same time as defined in the following subsections. We let $n$ and $\ell$ be the total number of the recipients and the maxium number of recipients in a ciphertext, respectively.

#### 3.4.1 *Semi-static Secure IBBE.*

- **Init**: The adversary $\mathcal{A}$ first declares a set of identities $S^* \subseteq \{1, ..., n\}$, which it wants to attack (with $|S^*| \leq \ell$), and we let $k = |S^*|$.
- **Setup**: The challenger runs **Setup**($\lambda$, $\ell$) to obtain a public key $PK$. It gives $\mathcal{A}$ the public key $PK$.
- **Phase I**: The adversary $\mathcal{A}$ issues queries for $ID_1, ..., ID_{k_1}$ to the challenger, where $ID_i$ is not in $S^*$ and the challenger forwards the resulting private keys to the adversary.
- **Challenge**: If **Phase I** is over, the challenger runs **Enc** algorithm to obtain $(Hdr^*, K)$ = **Enc**$(\tilde{S}$, PK) where $K \in \mathcal{K}$, and any $\tilde{S} \subseteq S^*$. The challenger set $K_0 = K$, and $K_1$ to a random value in $\mathcal{K}$, then randomly selects $b \leftarrow \{0, 1\}$. The challenger returns $(Hdr^*, K_b)$ to $\mathcal{A}$.
- **Phase II**: The adversary $\mathcal{A}$ continues to issue queries for $ID_{k_1+1}$, ..., $ID_{n-k}$ to the challenger, where $ID_i$ is not in $S^*$ and forwards the resulting private keys to the adversary.
- **Guess**: Finally, the adversary $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$ and wins the game if $b = b'$.

We define the advantage of the adversary $\mathcal{A}$ to win in the semi-static game as following:

$$AdvIBBE^{Semi}_{\mathcal{A},n,\ell}(\lambda) = \left|\Pr[b = b'] - \frac{1}{2}\right|.$$

Note that the selective security model can be derived straightforwardly by setting $\tilde{S} = S^*$.

#### 3.4.2 *Selectively Secure IBE.*

- **Init**: The adversary $\mathcal{A}$ first declares an identity $ID^* \in \{1, ..., n\}$ that it wants to attack.
- **Setup**: The challenger runs **Setup**($\lambda$) to obtain a public key $PK$. It gives $\mathcal{A}$ the public key $PK$.
- **Phase I**: The adversary $\mathcal{A}$ adaptively issues queries $ID_1, ..., ID_{k_1}$ to the challenger where $ID_i \neq ID^*$ and it forwards the resulting private keys to the adversary.
- **Challenge**: If **Phase I** is over, the challenger runs **Enc** to obtain $(PaHdr^*, K)$ = **Enc**$(S^*, ID^*, PK)$ where $S^*$ is a set of identities such that $ID^* \in S^*$ and $K \in \mathcal{K}$. The challenger sets $K_0 = K$, and $K_1$ to a random value in $\mathcal{K}$, then randomly selects $b \leftarrow \{0, 1\}$. The challenger returns $(PaHdr^*, K_b)$ to $\mathcal{A}$.
- **Phase II**: The adversary $\mathcal{A}$ continues to issue queries $ID_{k_1+1}, ..., ID_{n-1}$ to the challenger where $ID_i \neq ID^*$ and it forwards the resulting private keys to the adversary.
- **Guess**: Finally, the adversary $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$ and wins the game if $b = b'$.

We define the advantage of the adversary $\mathcal{A}$ to win in the selective security game as following:

$$AdvIBE^{Sel}_{\mathcal{A},n}(\lambda) = \left|\Pr[b = b'] - \frac{1}{2}\right|.$$

#### 3.4.3 *Adaptive Security Models.*

We defined the semi-static security of IBBE and the selective security of IBE above. It is straightforward to derive the adaptive security models of IBBE and IBE from those definitions by removing **Init** phase in the model. It means that the adversary does not give any information about the target before it issues a challenge. We define the advantages of those models as $AdvIBBE^{Adapt}_{\mathcal{A},n,\ell}(\lambda)$ (for IBBE) and $AdvIBE^{Adapt}_{\mathcal{A},n}(\lambda)$ (for IBE).

## 4 SYSTEM ARCHITECTURE

### 4.1 Overview

Our system targets an edge computing system, which consists of cloud, edges and end devices as illusted in Figure 1:

- **Cloud** is a centralized authority storing the data and imposing an access control and encryption on data before it distributes the data to edges. The cloud service provider has a key generation function. When an end device purchases a service, the cloud creates each a key and sends it through a secure channel.
- **Edges** are the closest network node to the end devices. The edge nodes are honest-but-curious. The service provider of edges is different from the cloud. They work with the cloud through

an SLA (Service Level Agreement), but they want to know the content of the end device data for their own purpose. Edges are usually less powerful than the cloud. They have smaller storage space and slower processors, but they are more powerful than most of end devices, which will be explained later. Under the SLA, edges allow an end device to access the data when the device who requests the data is a valid recipient of the encrypted data. Particularly, the edge wants to avoid end devices to receive any data without directly requesting the data or by eavesdropping the data sent to the other users.

- **End Devices** are resource-constrained IoT devices such as sensors and mobiles. They usually have limited computation and storage capability and often battery-powered. Each end device has a unique identity and a key which is associated with its identity. This key is stored in a secure memory of the device. We assume that the end device interests in the other end devices' data. It also wants to receive the data without being logged by the edge or the cloud.

## 4.2 Threat Models

We assume that the data in edge computing are exposed to the following two threats.

- **Threat on Edge** These are threats on the "edge" layer in Figure 1. In edge computing, edges are serviced by a third party service provider, which cannot be fully trusted by the cloud service provider. The edge service provider may have interests on end devices' data. Moreover, the edge can collude with some end devices to access targeted end devices' data. We can safely assume this since some devices belonging to the edge provider can join in the system as end devices by purchasing a service from the cloud service provider.
- **Threats on End Device** These are threats on the "end device" layer in Figure 1. Data stored in the edge are shared by multiple end devices. When data is requested by an end device, two problems may arise. First, edge should authenticate the end device. Second, data is broadcasted to the edge. Many end devices may receive the data but only the authorized requester should access the data. This ensures that end devices accessing data are properly monitored by the edge (or the cloud) to prevent scenarios such as consuming digital contents or updating software without payment. Typical IBBE and the existing outsourced computation schemes cannot prevent from these threats. Therefore, these schemes require additional authentication and one-to-one encryption such as WPA2 between an edge and an end device.

Our IBBE-OPD resolves both threats. Particularly, in our solution, the initial ciphertext created by the cloud and stored in the edge node cannot be decrypted by anyone except the end devices that the cloud sets as the recipient. Then, the partial decryption which is computed by the edge revokes all other users from the initial ciphertext except the one who has requested the content. Since the recipient is unique and each receiver has a private key associated with its identity, it does not require any extra authentication except the possession of the private key. A one-to-one encryption channel between the edge and the end device is not needed since the data cannot be decrypted by anyone else, except the requester.

## 4.3 Reference Scenarios

In this section, we introduce some reference scenarios for IBBE-OPT.

*4.3.1 Software Updates and Upgrades.* Software updates and upgrades pose a great challenge for systems employing large amount and variety of IoT devices. In a typical scenario, the cloud services may be operated by IoT vendors who are willing to periodically update and upgrade their products to reduce the risks associated with the most recent vulnerabilities identified in their products. The Edges may belong to organizations responsible for operating or maintaining a system supporting critical infrastructures such as energy, transportation and water. These systems may employ a large number and variety of end devices. A vendor would not like Edges to identify the updates performed on their devices for commercial reasons. Similarly, Edges cannot permit a vendor to learn about topology in which their devices are employed. Our IBBE-OPD efficiently solves this problem, i.e., a cloud service can encrypt updates and upgrades for their product ranges and release them to edge server. End device requests software from an edge, and the edge partially decrypts the requested software and delivers it to the end device.

*4.3.2 Military Scenario.* In a combat zone, it is quite challenging to maintain a reliable communication between command control and mission units (e.g., a squad or a platoon) operating deep in hostile territory. A common practice is to use air and/or ground vehicles with communication equipment at the edge to relay the messages. In a typical scenario, the command control may send mission-related information to the unit through the edge. IBBE-OPD can efficiently remove access to information irrelevant to the involved parties (e.g., unit members and their location, and mission information), and this way reduce the risk that information may leak to enemy intelligence.

## 4.4 Reference Protocol

Our IBBE-OPD can be deployed in our reference scenarios. The end devices can efficiently decrypt the ciphertext and retrieve the data. Specifically, our solution defines the following flow between a cloud service (CS), an edge server (ES) and an end device (ED):

(1) CS creates a public key and a master secret key using **Setup**.
(2) ED requests its private key associated with its identity by joining the system. CS generates it by running **KeyGen** and returns the private key to ED using a secure channel.
(3) CS encrypts the data using **Enc** for multiple receivers and sends the encrypted data to ES.
(4) ED presents its identity to ES and requests data. ES partially decrypts the encrypted data for the supplied identity and sends the partially decrypted ciphertext to ED.
(5) ED receives the partially decrypted ciphertext from ES and it decrypts the ciphertext using its private key and finally recovers the data.

# 5 OUR CONSTRUCTIONS

## 5.1 Construction of $\Pi_{\text{Semi}}$

Let $\mathcal{G}(\lambda, n, \ell)$ be an algorithm that outputs suitable bilinear group parameters $\langle \mathbb{G}, \mathbb{G}_T, e \rangle$, where $\mathbb{G}$ and $\mathbb{G}_T$ are of order $p \geq n + \ell$, and $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. We construct our scheme $\Pi_{\text{Semi}}$ as follows:

- **Setup**$(n, \ell)$: The algorithm runs $\langle \mathbb{G}, \mathbb{G}_T, e \rangle \leftarrow \mathcal{G}(\lambda, n, \ell)$. It randomly selects $g_1, g_2 \xleftarrow{R} \mathbb{G}$ and $\alpha, \beta, \gamma \xleftarrow{R} \mathbb{Z}_p$. It sets $\hat{g}_1 \leftarrow g_1^{\beta}$ and $\hat{g}_2 \leftarrow g_2^{\beta}$. $PK$ contains a description of the parameters $n$ and $\ell$, along with $g_1^{\gamma}, g_1^{\gamma \cdot \alpha}$ and the set
$$\{g_1^{\alpha^i}, \hat{g}_1^{\alpha^i}, \hat{g}_2^{\alpha^j} : i \in [0, \ell], j \in [0, \ell - 2]\}.$$
The master secret key is $MSK \leftarrow (\alpha, \gamma, g_2)$.
- **KeyGen**(ID, $MSK$): For each user's identity ID, it outputs the private key
$$sk_{\text{ID}} \leftarrow g_2^{\frac{\gamma}{\alpha - \text{ID}}}.$$
- **Enc**$(S, PK)$: For a set of identities $S$, we let $k = |S|$. The algorithm parses $S$ as $\{s_1, ..., s_k\}$. It randomly selects $t \xleftarrow{R} \mathbb{Z}_p$ and sets $K \leftarrow e(g_1, \hat{g}_2)^{\gamma \cdot \alpha^{\ell-1} \cdot t}$. It sets $C_1 = \hat{g}_1^{P(S,\alpha) \cdot t}$ and $C_2 = g_1^{\gamma \cdot t}$ where $P(S, x) = x^{\ell - k} \prod_{j=1}^{k} (x - s_j)$. It outputs
$$Hdr = \langle C_1, C_2 \rangle \text{ and } K.$$
- **PaDec**$(S, \text{ID}, Hdr, PK)$: Suppose $\text{ID} \in S$. Let $Hdr = \langle C_1, C_2 \rangle$ and $P_{ID}(S, x) = x^{\ell-1} - \frac{P(S,x)}{x - ID}$. It sets $C_1' = C_1$ and $C_2' = e(C_2, \hat{g}_2^{P_{ID}(S,\alpha)})$. It outputs
$$PaHdr = \langle C_1', C_2' \rangle.$$
- **CoDec**$(sk_{ID}, PaHdr)$: Parse $PaHdr$ to $\langle C_1', C_2' \rangle$. It outputs
$$K = e(C_1', sk_{ID}) \cdot C_2'.$$

**Correctness**. The correctness of our scheme is shown as follows:
$$\begin{aligned} K^{1/t} &= e(\hat{g}_1^{P(S,\alpha)}, g_2{}^{\gamma/(\alpha - \text{ID})}) \cdot e(g_1{}^{\gamma}, \hat{g}_2^{P_{ID}(S,\alpha)}) \\ &= e(g_1, \hat{g}_2)^{\gamma(P(S,\alpha)/(\alpha - \text{ID}) + P_{ID}(S,\alpha))} \\ &= e(g_1, \hat{g}_2)^{\gamma \cdot \alpha^{\ell-1}}. \end{aligned}$$

## 5.2 Construction of $\Pi_{\text{Adapt}}$

Let $\mathcal{G}(\lambda, n, \ell)$ be an algorithm that outputs suitable bilinear group parameters $\langle \mathbb{G}, \mathbb{G}_T, e \rangle$, where $\mathbb{G}$ and $\mathbb{G}_T$ are of order $p \geq 2n + \ell$. Using $\Pi_{\text{Semi}} = \{ \textbf{Setup}_{\text{Semi}}, \textbf{KeyGen}_{\text{Semi}}, \textbf{Enc}_{\text{Semi}}, \textbf{Dec}_{\text{Semi}} \}$, we construct a new adaptively secure construction $\Pi_{\text{Adapt}}$ as follows:

- **Setup**$(n, \ell)$: The algorithm runs $\textbf{Setup}_{\text{Semi}}(2n, \ell)$ to get $PK', MSK'$. It sets $PK \leftarrow PK'$ and $MSK \leftarrow MSK'$.
- **KeyGen**(ID, $MSK$): For an identity ID, the algorithm randomly selects a bit $v_{\text{ID}}$. It runs $\textbf{KeyGen}_{\text{Semi}}(2 \cdot \text{ID} + v_{\text{ID}}, MSK')$ to get $sk'_{\text{ID}}$ and set $sk_{\text{ID}} = (sk'_{ID}, v_{ID})$.
- **Enc**$(S, PK)$: For a set of identities $S$, we use $s_i$ to denote the $i$th identity in $S$. The algorithm generates an $|S|$ bits random value $u$ where $u_i$ denotes the $i$th bit of $u$. It sets
$$S_0 := \{2s_i + u_i : i \in |S|\} \text{ and } S_1 := \{2s_i + 1 - u_i : i \in |S|\}$$
It computes $\langle Hdr_0, k_0 \rangle$ and $\langle Hdr_1, k_1 \rangle$ as follows:
$$\langle Hdr_0, k_0 \rangle \leftarrow \textbf{Enc}_{\text{Semi}}(S_0, PK'),$$
$$\langle Hdr_1, k_1 \rangle \leftarrow \textbf{Enc}_{\text{Semi}}(S_1, PK')$$
It randomly generates $K \in \mathcal{K}$. Using symmetric encryption algorithm $SymEnc(symkey, M)$ for a symmetric encryption key $symkey$ and a message $M$, it computes $K_0 = SymEnc(k_0, K)$ and $K_1 = SymEnc(k_1, K)$. It outputs $K$ and a ciphertext
$$CT := \langle Hdr_0, K_0, Hdr_1, K_1, u \rangle.$$
- **PaDec**$(S, ID, CT, PK)$: Suppose $ID \in S$. The algorithm parses $CT$ to $Hdr_0, K_0, Hdr_1, K_1$ and $u$. It computes $S_0$ and $S_1$ using $S$ and $u$. It computes the followings:
$$PaHdr_0 \leftarrow \textbf{PaDec}_{\text{Semi}}(S_0, 2 \cdot ID + u_{ID}, Hdr_0, PK')$$
$$PaHdr_1 \leftarrow \textbf{PaDec}_{\text{Semi}}(S_1, 2 \cdot ID + 1 - u_{ID}, Hdr_1, PK')$$
where $u_{ID}$ is the bit allocated to ID in $u$. It outputs
$$PaCT := \langle PaHdr_0, K_0, PaHdr_1, K_1, u_{ID} \rangle.$$
- **CoDec**$(sk_{\text{ID}}, PaCT)$: It parses $PaCT$ to $PaHdr_0, K_0, PaHdr_1, K_1$ and $u_{\text{ID}}$ and $sk_{\text{ID}}$ to $sk'_{\text{ID}}$ and $v_{\text{ID}}$. It computes
$$k_{u_{\text{ID}} \oplus v_{\text{ID}}} = \textbf{CoDec}_{\text{Semi}}(d'_i, PaHdr_{u_{\text{ID}} \oplus v_{\text{ID}}}).$$
Using $k_{u_{\text{ID}} \oplus v_{\text{ID}}}$, it runs either $SymDec(k_0, K_0)$ or $SymDec(k_1, K_1)$ to provide $K$ where $SymDec$ is the decryption algorithm to match to $SymEnc$.

**Correctness**. In this scheme, we set the total number of users as $2n$. We allow a user to decrypt ciphertext using $S_0$ and $S_1$, whether its $u_{ID} \oplus v_{ID}$ is 0 or 1. The decryption can be permitted using $\textbf{PaDec}_{\text{Semi}}$ and $\textbf{CoDec}_{\text{Semi}}$ of $\Pi_{\text{Semi}}$ by applying those algorithms to $S_0$ and $S_1$.

**Table 1: Algorithm Map between IBBE-OPD, IBBE and IBE**

| IBBE-OPD ($\Pi$) | IBBE ($\Pi^{\text{ibbe}}$) | IBE ($\Pi^{\text{ibe}}$) |
|---|---|---|
| **Setup** | $\textbf{Setup}^{\text{ibbe}}$ | $\textbf{Setup}^{\text{ibe}}$ |
| **KeyGen** | $\textbf{KeyGen}^{\text{ibbe}}$ | $\textbf{KeyGen}^{\text{ibe}}$ |
| **Enc** | $\textbf{Enc}^{\text{ibbe}}$ | $\textbf{Enc}^{\text{ibe}}$ |
| **PaDec** | $\textbf{Dec}^{\text{ibbe}}$ | |
| **CoDec** | | $\textbf{Dec}^{\text{ibe}}$ |

# 6 SECURITY ANALYSIS

We prove the security of our schemes based on the threats that we have defined in the previous section. Our schemes protect the data from two different adversaries. Particularly, if an adversary (e.g., an edge in our system description) wants to break the ciphertext received directly from the cloud server before the partial decryption algorithm is applied, the adversary is a typical IBBE adversary. We can prove the security by forming an IBBE from our IBBE-OPT by merging **PaDec** and **CoDec** to the decryption algorithm $\textbf{Dec}^{\text{ibbe}}$ of IBBE. On the other hand, if an adversary (e.g., end devices in our system description) wants to break the ciphertext which is partially decrypted by an edge, the adversary is an IBE adversary. We can prove the security of this adversary by deriving the encryption algorithm $\textbf{Enc}^{\text{ibe}}$ of IBE by merging **Enc** and **PaDec**. We provide summary of the algorithm mapping in Table 1.

## 6.1 IBBE Construction of $\Pi_{\mathbf{Semi}}$

The first scheme, which can be derived from our semi-static scheme $\Pi_{\text{Semi}}$ is an IBBE scheme $\Pi^{\text{ibbe}}_{\text{Semi}}$ which is identical to JWMJ broadcast encryption scheme [12]. Therefore, we can construct an IBBE scheme $\Pi^{\text{ibbe}}_{\text{Semi}} = \{\mathbf{Setup}^{\text{ibbe}}_{\text{Semi}}, \mathbf{KeyGen}^{\text{ibbe}}_{\text{Semi}}, \mathbf{Enc}^{\text{ibbe}}_{\text{Semi}}, \mathbf{Dec}^{\text{ibbe}}_{\text{Semi}}\}$ by setting $\mathbf{Setup}^{\text{ibbe}}_{\text{Semi}}$, $\mathbf{KeyGen}^{\text{ibbe}}_{\text{Semi}}$ and $\mathbf{Enc}^{\text{ibbe}}_{\text{Semi}}$ identical with **Setup**, **KeyGen** and **Enc** of the construction $\Pi_{Semi}$ (Section 5.1), respectively. We define $\mathbf{Dec}^{\text{ibbe}}_{\text{Semi}}$ as follows:

- $\mathbf{Dec}^{\text{ibbe}}_{\text{Semi}}(S, ID, sk_{ID}, Hdr, PK)$: Suppose $ID \in S$. Let $Hdr = \langle C_1, C_2\rangle$ and $P_{ID}(S,x) = x^{\ell-1} - \frac{P(S,x)}{x-ID}$. It outputs
$$K = e(C_1, sk_{ID}) \cdot e(C_2, \hat{g}_2^{P_{ID}(S,\alpha)}).$$

Theorem 6.1. *IBBE scheme described above is semi-static secure.*

Proof. The security of $\Pi^{\text{ibbe}}_{\text{Semi}}$ is proven by JWMJ in [12]. The scheme is a semi-static secure under *DB-DHES* assumption. □

## 6.2 IBE Construction of $\Pi_{\mathbf{Semi}}$

The second scheme, which can be derived from $\Pi_{\text{Semi}}$, is an IBE scheme $\Pi^{\text{ibe}}_{\text{Semi}}$*. In this scheme, the ciphertext can be decrypted by only one recipient, not a group of recipients. We define $\Pi^{\text{ibe}}_{\text{Semi}} = \{\mathbf{Setup}^{\text{ibe}}_{\text{Semi}}, \mathbf{KeyGen}^{\text{ibe}}_{\text{Semi}}, \mathbf{Enc}^{\text{ibe}}_{\text{Semi}}, \mathbf{Dec}^{\text{ibe}}_{\text{Semi}}\}$ where $\mathbf{Setup}^{\text{ibe}}_{\text{Semi}}$, $\mathbf{KeyGen}^{\text{ibe}}_{\text{Semi}}$ and $\mathbf{Dec}^{\text{ibe}}_{\text{Semi}}$ are identical with **Setup**, **KeyGen** and **CoDec** of the construction of $\Pi_{\text{Semi}}$ (Section 5.1), respectively. $\mathbf{Enc}^{\text{ibe}}_{\text{Semi}}$ is defined by combining **Enc** and **PaDec** as follows:

- $\mathbf{Enc}^{\text{ibe}}_{\text{Semi}}(S, ID, PK)$: For a set of identities $S$ and an identity $ID \in S$. We let $k = |S|$. The algorithm parses $S$ as $\{s_1, ..., s_k\}$†. It randomly selects $t \xleftarrow{R} \mathbb{Z}_p$ and set $K \leftarrow e(g_1, \hat{g}_2)^{\gamma\cdot\alpha^{\ell-1}\cdot t}$. It sets $C'_1 = \hat{g}_1^{P(S,\alpha)\cdot t}$ and $C'_2 = e(g_1^{\gamma\cdot t}, \hat{g}_2^{P_{ID}(S,\alpha)})$ where $P(S,x) = x^{\ell-k}\prod_{j=1}^{\ell}(x-s_j)$ and $P_{ID}(S,x) = x^{\ell-1} - \frac{P(S,x)}{x-ID}$. It outputs
$$PaHdr = \langle C'_1, C'_2\rangle.$$

**Remark 1**. In the above IBE, the set of recipients $S$ seems to be redundant. It is only kept as a parameter in the construction for the consistency with the original scheme. But, it should be noted that $S$ must contain the recipient's identity $ID$ as an element for the decryption.

Theorem 6.2. *The IBE scheme $\Pi^{ibe}_{Semi}$ is selectively secure.*

Proof. We assume that DB-DHES instance $\{g^{\alpha^i} : i \in Z\}$ is given with $m = 4d + 4\ell - 1$ for

$$Z = [0, \ell-2] \cup [d+\ell, 2d+\ell-1] \cup [2d+2\ell, 2d+3\ell-1] \cup$$
$$[3d+3\ell, 4d+3\ell] \cup [4d+4\ell, 5d+4\ell+1]$$

where $d = n + 2\ell$.

**Init** $\mathcal{A}$ selects $ID^* \in [1, n]$ and sends $ID^*$ to $\mathcal{B}$.

**Setup** $\mathcal{B}$ randomly selects $a_0, a_1, a_2 \xleftarrow{R} \mathbb{Z}_p^*$ and sets

$$f(x) = \prod_{i\in[1,n]\setminus\{ID^*\}} (x-i)\cdot f'(x).$$

where $f'(x)$ is a $2\ell+1$ degree polynomial not to have roots in $[0, n]$. It should be noted that $f(x)$ is a $d$ degree polynomial. Then, $\mathcal{B}$ sets

$$\beta \leftarrow a_0 \cdot \alpha^{-d-\ell},\ \gamma \leftarrow f(\alpha),\ \tilde{g}_1 \leftarrow g^{a_1},\ \tilde{g}_2 \leftarrow g^{a_2}$$

and

$$g_1 \leftarrow \tilde{g}_1^{\alpha^{4d+4\ell}},\ g_2 \leftarrow \tilde{g}_2^{\alpha^{d+\ell}},\ \hat{g}_1 \leftarrow g_1^{\beta},\ \hat{g}_2 \leftarrow g_2^{\beta}.$$

Finally, $\mathcal{B}$ computes the public key

$$PK = \{g_1^{\gamma}, g_1^{\gamma\cdot\alpha}, g_1^{\alpha^i}, \hat{g}_1^{\alpha^i}, \hat{g}_2^{\alpha^j} : i \in [0,\ell], j \in [0, \ell-2]\}$$

and sends $PK$ to $\mathcal{A}$.

**Phase I/II** If $\mathcal{A}$ makes a private key query for $i \in [1, n] \setminus \{ID^*\}$, $\mathcal{B}$ sets

$$sk_i = g_2^{\frac{\gamma}{\alpha-i}} = g_2^{\frac{f(\alpha)}{\alpha-i}} = g^{a_2\alpha^{d-\ell}\cdot\frac{f(\alpha)}{\alpha-i}}.$$

Because $i$ is a root of $f(x)$ for all $i \in [1, n]\setminus\{ID^*\}$, $\frac{f(\alpha)}{\alpha-i}$ is $d-1$ degree polynomial and $\mathcal{B}$ can compute $sk_i$ using $\{g^{\alpha^i} : i \in [d+\ell, 2d+\ell-1]\}$, which are given in the instance.

**Challenge** For simplicity, we let $g_3^{\alpha^{d+\ell}} = g_1$, and $\hat{g}_3^{\alpha^{d+\ell}} = \hat{g}_1$ (i.e., $g_3 = \tilde{g}_1^{\alpha^{3d+3\ell}}$, $\hat{g}_3 = \tilde{g}_1^{a_0\cdot\alpha^{2d+2\ell}}$). Then, $g_3$ and $\hat{g}_3$ can be computed only when they are in the following set:

$$\{g_3^{\alpha^i}, \hat{g}_3^{\alpha^j} : i \in [0,d] \cup [d+\ell, 2d+\ell+1],$$
$$j \in [0, \ell-1] \cup [d+\ell, 2d+\ell] \cup [2d+2\ell, 3d+2\ell+1]\}.$$

We, then, implicitly set $t = \alpha^{-d-\ell}\cdot t'(\alpha)$ since $t$ appears both in $C'_1$ and $C'_2$.

We let $f(x)|_i$ denote the coefficient of $x^i$ in function $f$ and set $k = |S|$. If $\mathcal{A}$ sends a request of the challenge ciphertext for $ID^*$ with the set of recipients $S$ such that $ID^* \in S$, $\mathcal{B}$ computes a polynomial $\tilde{f}(x)$ such that $f(x) = \tilde{f}(x)\cdot x^{-(\ell-k)}\cdot(x^{\ell-1} - P_{ID^*}(S,x))$. It is worth noting that $\tilde{f}(x)$, which is $d-k+1$ degree polynomial, does not have any common root with $P(S,x)$ (i.e., $\tilde{f}(x)$ and $P(S,x)$ are redundant). Therefore, by the Lemma 1 of [7], there exists $t'(x)$, of which degree is $d+2\ell-k-1$, such that

$$t'(x)\tilde{f}(x)|_i = 0,\ \text{if } i \in [d-k+2, d+2\ell-k-1],$$

$$t'(x)\tilde{f}(x)|_{d-k+1} = 1 \text{ and } t'(x)P(S,x)|_i = 0,\ \text{if } i \in [\ell, d+2\ell-k-1]$$

$\mathcal{B}$ now can compute

$$C'_1 = \hat{g}_3^{t'(\alpha)\cdot P(S,\alpha)} \text{ and } C'_2 = e(g_3, \hat{g}_2)^{t'(x)f(x)P_{ID*}(S,\alpha)}.$$

In particular, $C'_2$ can be computed using the given instance because

$$e(g_3, \hat{g}_2)^{t'(\alpha)f(\alpha)P_{ID*}(S,\alpha)}$$
$$= e(g,g)^{a_0a_1a_2\alpha^{3d+3\ell}(t'(\alpha)\tilde{f}(\alpha)\cdot\alpha^{-(\ell-k)}\cdot(\alpha^{\ell-1}-P_{ID*}(S,\alpha))P_{ID*}(\alpha)}$$
$$= e(g,g)^{a_0a_1a_2\alpha^{3d+2\ell+k}(t'(\alpha)\tilde{f}(\alpha)(\alpha^{\ell-1}-P_{ID*}(S,\alpha))P_{ID*}(\alpha)}$$

*This scheme is selectively secure, but we keep the subscript "Semi" in the notation for the consistency.

†There exists $j \in [1, k]$ such that $s_j = ID$.

To show that $C_2'$, this is computable, we split $t'(\alpha)\tilde{f}(\alpha)$ into two polynomials $A(\alpha)$ and $B(\alpha)$ such that

$$t'(\alpha)\tilde{f}(\alpha) = \underbrace{c_0 + c_1\alpha + \ldots + c_{d-\ell}\alpha^{d-k} + \alpha^{d-k+1}}_{A(\alpha)} + \underbrace{c_{d+2\ell-k}\alpha^{d+2\ell-k} + \ldots + c_{2d+\ell}\alpha^{2d+\ell}}_{B(\alpha)}.$$

Therefore, $t'(\alpha)\tilde{f}(\alpha) = A(x) + B(x)$. This implies that the highest degree of monomials in $A(x)$ is $d-k+1$ and the lowest degree of monomials in $B(x)$ is $d+2\ell-k$. Because $P_{ID^*}(S,x)$ is a $\ell-2$ degree polynomial, $(\alpha^{\ell-1} - P_{ID^*}(S,\alpha))P_{ID^*}(S,\alpha)$ is a $2\ell-3$ degree polynomial. Therefore, the highest degree of monomials in $A(x)(\alpha^{\ell-1} - P_{ID^*}(S,\alpha))P_{ID^*}(S,\alpha)$ is $d+2\ell-k-2$. The lowest degree of monomials in $B(x)(\alpha^{\ell-1} - P_{ID^*}(S,\alpha))P_{ID^*}(S,\alpha)$ remains $d+2\ell-k$.

It implies that ,

$$\alpha^{3d+2\ell+k} \cdot t(\alpha)\tilde{f}(\alpha)(\alpha^{\ell-1} - P_{ID^*}(S,\alpha))P_{ID^*}(S,\alpha)|_{4d+4\ell-1} = \alpha^{3d+2\ell+k} \cdot t(\alpha)f(\alpha)P_{ID^*}(S,\alpha)|_{4d+4\ell-1} = 0$$

and computing $C_2'$ does not involve $e(g,g)^{4d+4\ell-1}$, which cannot be computable.

Also, $K$ can be computed efficiently by setting

$$K \leftarrow T^{a_0a_1a_2} \cdot e(g,g)^{a_0a_1a_2(f(\alpha)\cdot t(\alpha)\cdot\alpha^{3d+4\ell-1} - \alpha^{4d+4\ell-1})}.$$

If $T = e(g_1,g_2)^{\alpha^{4d+4\ell-1}}$, then $K$ is valid. Otherwise, the random will be added to $K$.

**Guess** Finally, $\mathcal{A}$ outputs a bit $b'$. $\mathcal{B}$ sends $b'$ to the challenger.

□

## 6.3 IBBE Construction of $\Pi_{\text{Adapt}}$

We also can derive an IBBE scheme $\Pi^{\text{ibbe}}_{\text{Adapt}}$ = {**Setup**$^{\text{ibbe}}_{\text{Adapt}}$, **KeyGen**$^{\text{ibbe}}_{\text{Adapt}}$, **Enc**$^{\text{ibbe}}_{\text{Adapt}}$, **Dec**$^{\text{ibbe}}_{\text{Adapt}}$} from $\Pi_{\text{Adapt}}$ = { **Setup**$_{\text{Adapt}}$, **KeyGen**$_{\text{Adapt}}$, **Enc**$_{\text{Adapt}}$, **Dec**$_{\text{Adapt}}$} which is almost identical to JWMJ's adaptively secure broadcast encryption scheme [12]. The difference is that the public key generator (PKG) running **Setup** algorithm does not need to keep $n$ bits in the master secret key where $n$ is the total number of users. These bits can be created on the fly when PKG generates a private key for a user in **KeyGen**. We observed that these bits are not necessarily defined in **Setup** since they were defined as a master secret which is never given to the adversary in the proof. Although they are all chosen in **Setup**, the adversary will not know whether they are selected in **Setup** or when a new private is queried. Therefore, including these values in a master secret key is redundant and consumes the secure memory of PKG.

We can construct $\Pi^{\text{ibbe}}_{\text{Adapt}}$ from **Setup**, **KeyGen** and **Enc** of $\Pi_{\text{Adapt}}$ (Section 5.2), respectively. **Dec**$^{\text{ibbe}}_{\text{Adapt}}$ is defined as follows:

- **Dec**$^{\text{ibbe}}_{\text{Adapt}}$($S$, $ID$, $sk_{ID}$, $CT$, $PK$): Suppose $ID \in S$. The algorithm parses $CT$ to $Hdr_0, K_0, Hdr_1, K_1$ and $u$ and $sk_{ID}$ to $sk'_{ID}$ and $v_{ID}$. It computes $S_0$ and $S_1$ using $S$ and $u$. We let $u_{ID}$ be the bit associated to $ID$ in $u$. It, then, computes one of the followings:

  If $u_{\text{ID}} \oplus v_{\text{ID}} = 0$,

  $PaHdr_0 \leftarrow$ **PaDec**$_{\text{Semi}}(S_0, 2\cdot ID + u_{ID}, Hdr_0, PK')$.

  If $u_{\text{ID}} \oplus v_{\text{ID}} = 1$,

  $PaHdr_1 \leftarrow$ **PaDec**$_{\text{Semi}}(S_1, 2\cdot ID + 1 - u_{ID}, Hdr_1, PK')$.

  It computes $k_{u_{\text{ID}}\oplus v_{\text{ID}}} =$ **CoDec**$_{\text{Semi}}(sk'_i, PaHdr_{u_{\text{ID}}\oplus v_{\text{ID}}})$. Then, it decrypts $K_{u_{\text{ID}}\oplus v_{\text{ID}}}$ using $k_{u_{\text{ID}}\oplus v_{\text{ID}}}$ (i.e., it computes either $SymDec(k_0, K_0)$ or $SymDec(k_1, K_1)$) to get $K$.

Theorem 6.3. $\Pi^{ibbe}_{Adapt}$ *is adaptively secure.*

Proof. The security of $\Pi^{\text{ibbe}}_{\text{Adapt}}$ is proven by JWMJ's adaptive scheme. The scheme is a adaptively secure under *DB-DHES* assumption. □

## 6.4 IBE Construction of $\Pi_{\text{Adapt}}$

The second scheme, which can be derived from our $\Pi_{\text{Adapt}}$, is an adaptively secure IBE scheme $\Pi^{ibe}_{Adapt}$. In this scheme, the ciphertext can be decrypted only for a receiver, not a group of receivers. We define $\Pi^{\text{ibe}}_{\text{Adapt}}$ = {**Setup**$^{\text{ibe}}_{\text{Adapt}}$, **KeyGen**$^{\text{ibe}}_{\text{Adapt}}$, **Enc**$^{\text{ibe}}_{\text{Adapt}}$, **Dec**$^{\text{ibe}}_{\text{Adapt}}$} where **Setup**$^{\text{ibe}}_{\text{Adapt}}$, **KeyGen**$^{\text{ibe}}_{\text{Adapt}}$ and **Dec**$^{\text{ibe}}_{\text{Adapt}}$ are identical with **Setup**, **KeyGen** and **CoDec** (Section 5.1), respectively. **Enc**$^{\text{ibe}}_{\text{Adapt}}$ is defined by combining **Enc** and **PaDec** as follows:

- **Enc**($S$, $ID$, $PK$): For a set of identities $S$, we use $s_i$ to denote the $i$th user in $S$. The algorithm generates a $|S|$ bits random value $u$ and we also use $u_i$ to denote the $i$th bit of $u$. It sets

  $S_0 := \{2s_i + u_i : i \in |S|\}$ and $S_1 := \{2s_i + (1-u_i) : i \in |S|\}$

  It computes $\langle Hdr_0, k_0\rangle$ and $\langle Hdr_1, k_1\rangle$ as follows:

  $\langle Hdr_0, k_0\rangle \leftarrow$ **Enc**$'(S_0, PK')$ and $\langle Hdr_1, k_1\rangle \leftarrow$ **Enc**$'(S_1, PK')$

  It randomly generates $K \in \mathcal{K}$. Using symmetric encryption algorithm $Sym$. It computes $K_0 = Sym(k_0, K)$ and $K_1 = Sym(k_1, K)$. Then, for $ID \in S$, the algorithm computes the followings:

  $PaHdr_0 \leftarrow$ **PaDec**$'_{\text{Adapt}}(S_0, 2ID + u_{ID}, Hdr_0, PK')$

  $PaHdr_1 \leftarrow$ **PaDec**$'_{\text{Adapt}}(S_1, 2ID + (1-u_{ID}), Hdr_1, PK')$

  where $u_{ID}$ is the bit associated to $ID$ in $u$. It outputs $PaCT := \langle PaHdr_0, K_0, PaHdr_1, K_1, u_{ID}\rangle$.

Theorem 6.4. *IBE scheme described above is adaptively secure.*

Proof. The security of the IBE scheme is proved by Lemmas 1 and 2. □

To prove the security of $\Pi^{\text{ibe}}_{\text{Adapt}}$, we define the following security games:

- **Game**$_0$: This is a real game simulating adaptive security using $\Pi^{\text{ibe}}_{\text{Adapt}}$.
- **Game**$_1$: This is identical to **Game**$_0$ except that a random key replace $k_0$.
- **Game**$_2$: This is identical to **Game**$_1$ except that a random key replace $k_1$.

- **Game**$_3$: This is identical to **Game**$_2$ except that a random key replace $K$ in $K_0$.
- **Game**$_4$: This is identical to **Game**$_3$ except that a random key replace $K$ in $K_1$.

We prove that **Game**$_0$, **Game**$_1$ and **Game**$_2$ are indistinguishable in Lemmas 1 and 2. After both $k_0$ and $k_1$ are replaced by random keys, the ciphertext does not include any information of the key encrypts $SymEnc(k_0, K)$ and $SymEnc(k_1, K)$. Therefore, **Game**$_2$, **Game**$_3$ and **Game**$_4$ are indistinguishable due to the security of the symmetric encryption algorithm.

**Lemma 1.** *Suppose there exists a PPT algorithm $\mathcal{A}$, which can distinguish between **Game**$_0$ and **Game**$_1$ which are indistinguishable with non-negligible advantage $\epsilon$. Then, one can build an algorithm $\mathcal{B}$ which breaks DB-DHES assumption using $\mathcal{A}$ with $\epsilon$.*

Proof. Let us assume that DB-DHES instance $\{T, g^{\alpha^i} : i \in Z\}$ is given with $m = 4d + 4\ell - 1$ for

$$Z = [0, \ell - 2] \cup [d + \ell, 2d + \ell - 1] \cup [2d + 2\ell, 2d + 3\ell - 1] \cup$$
$$[3d + 3\ell, 4d + 3\ell] \cup [4d + 4\ell, 5d + 4\ell + 1],$$

where $\tilde{g}$ is generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively, and $d = n + 2\ell$. Then, $\mathcal{B}$ will break DB-DHES assumption by distinguishing whether $T$ is $e(g, g)^m$ or a random from $\mathbb{G}_T$ using an algorithm $\mathcal{A}$.

**Setup** $\mathcal{B}$ randomly selects $a_0, a_1, a_2 \xleftarrow{R} \mathbb{Z}_p^*$ and $n$ random bits $v$ from $\{0, 1\}^n$. It sets

$$f(x) = \prod_{i \in [1,n]} (x - (2i + v_i)) \cdot f'(x).$$

where $f'(x)$ is a $2\ell$ degree polynomial not having roots in $[0, 2n]$. It should be noted that $f(x)$ is a $d$ degree polynomial. Then, $\mathcal{B}$ sets

$$\beta \leftarrow a_0 \cdot \alpha^{-d-\ell},\ \gamma \leftarrow f(\alpha),\ \tilde{g}_1 \leftarrow g^{a_1},\ \tilde{g}_2 \leftarrow g^{a_2},$$

and

$$g_1 \leftarrow \tilde{g}_1^{\alpha^{4d+4\ell}},\ g_2 \leftarrow \tilde{g}_2^{\alpha^{d+\ell}},\ \hat{g}_1 \leftarrow g_1^{\beta},\ \hat{g}_2 \leftarrow g_2^{\beta}.$$

Finally, $\mathcal{B}$ computes the public key

$$PK = \{g_1^{\gamma}, g_1^{\gamma \cdot \alpha}, g_1^{\alpha^i}, \hat{g}_1^{\alpha^i}, \hat{g}_2^{\alpha^j} : i \in [0, \ell], j \in [0, \ell - 2]\}$$

and sends $PK$ to $\mathcal{A}$.

**Phase I/II** If $\mathcal{A}$ makes a private key query for $i \in [1, n]$, $\mathcal{B}$ sets

$$sk_i = g_2^{\frac{\gamma}{\alpha - (2i + v_i)}} = g_2^{\frac{f(\alpha)}{\alpha - (2i + v_i)}} = g^{a_2 \alpha^{d+\ell} \cdot \frac{f(\alpha)}{\alpha - (2i + v_i)}}.$$

Because $(2i + v_i)$ is a root of $f(x)$ for all $i \in [1, n]$, $\frac{f(\alpha)}{\alpha - (2i + v_i)}$ is $d - 1$ degree polynomial and $\mathcal{B}$ can compute $d_i$ using $\{g^{\alpha^i} : i \in [d + \ell, 2d + \ell - 1]\}$, which are given in the instance. $\mathcal{B}$ sends $\{sk_i, v_i\}$ to $\mathcal{A}$.

**Challenge** For simplicity, we let $g_3^{\alpha^{d+\ell}} = g_1$, and $\hat{g}_3^{\alpha^{d+\ell}} = \hat{g}_1$ (i.e., $g_3 = \tilde{g}_1^{\alpha^{3d+3\ell}}$, $\hat{g}_3 = \tilde{g}_1^{a_0 \cdot \alpha^{2d+2\ell}}$). Then, $g_3$ and $\hat{g}_3$ can be computed only when they are in the following set:

$$\{g_3^{\alpha^i}, \hat{g}_3^{\alpha^j} : i \in [0, d] \cup [d + \ell, 2d + \ell + 1],$$
$$j \in [0, \ell - 1] \cup [d + \ell, 2d + \ell] \cup [2d + 2\ell, 3d + 2\ell + 1]\}.$$

We, then, implicitly set $t = \alpha^{-d-\ell} \cdot t'(\alpha)$ since $t$ appears both in $C_1'$ and $C_2'$. We let $f(x)|_i$ denote the coefficient of $x^i$ in function $f$ . If $\mathcal{A}$ sends a request of the challenge ciphertext for $ID^*$ with the a set of recipients $S$ such that $ID^* \in S$, $\mathcal{B}$ randomly selects $\ell$ random bits $u$ and sets $u_{ID^*} = (1 - v_{ID^*})$ where $u_{ID^*}$ and $v_{ID^*}$ are bits corresponding to $ID^*$ in $u$ and $v$. It should be noted that $u_{ID^*}$ is still random to the adversary since $v_{ID^*}$ are never queried before. We set $S_0 = \{2s_i + u_i : i \in |S|\}$ and $k = |S_0|(= |S|)$ then it computes $P(S_0, x)$.

$\mathcal{B}$ computes a polynomial $\tilde{f}(x)$ such that $\tilde{f}(x) \cdot x^{-(\ell-k)}(x^{\ell-1} - P_{ID^*}(S_0, x))$. $\tilde{f}(x)$ has $d-k+1$ degree and does not have any common root with with $P(S_0, x)$ (i.e., $\tilde{f}(x)$ and $P(S_0, x)$ are redundant). $\mathcal{B}$ also computes a polynomial $t'(x)$ of degree $d + 2\ell - k - 1$ such that

$$t'(x)\tilde{f}(x)|_i = 0, \text{ if } i \in [d-k+2, d+2\ell-k-1], \quad t'(x)\tilde{f}(x)|_{d-k+1} = 1$$
$$\text{and } t'(x)P(S_0, x)|_i = 0, \text{ if } i \in [\ell, d + 2\ell - k - 1].$$

By the lemma 1 of [7], there exists $t'(x)$ satisfying the conditions above.

$\mathcal{B}$ now can compute

$$C_1' = \hat{g}_3^{t'(\alpha) \cdot P(S_0, \alpha)} \text{ and } C_2' = e(g_3, \tilde{g}_2)^{t'(\alpha) f(\alpha) P_{ID^*}(\alpha)}.$$

In particular, $C_2'$ can be computed using the given instances. Since

$$e(g_3, \tilde{g}_2)^{t'(\alpha) f(\alpha) P_{ID^*}(S_0, \alpha)}$$
$$= e(g, g)^{a_0 a_1 a_2 \alpha^{3d+2\ell+k}(t'(\alpha)\tilde{f}(\alpha)(\alpha^{\ell-1} - P_{ID^*}(S_0, \alpha)) P_{ID^*}(S_0, \alpha)}$$

$C_2'$ is efficiently computable for the similar reason we explained in Theorem 5.2. Also, $k_0$ can be computed efficiently by setting

$$k_0 \leftarrow T^{a_0 a_1 a_2} \cdot e(g, g)^{a_0 a_1 a_2 (f(\alpha) \cdot t(\alpha) \cdot \alpha^{3d+4\ell-1} - \alpha^{4d+4\ell-1})}.$$

$\mathcal{B}$ randomly selects $K \leftarrow \mathcal{K}$ and computes $K_0 = Sym(k_0, K)$. It also sets $PaHdr_0 = (C_1, C_2)$. Then, $\mathcal{B}$ runs $(Hdr_1, k_1) \leftarrow \mathbf{Enc}'(S_1, PK')$ and $PaHdr_1 \leftarrow PaDec'(S_1, 2 \cdot ID + (1 - u_{ID}), Hdr_1, PK')$. It sends $PaCT := \langle PaHdr_0, K_0, PaHdr_1, K_1, u_{ID} \rangle$ to $\mathcal{A}$.

If $T = e(g, g)^{\alpha^{4d+4\ell-1}}$, then $k_0$ is valid and this will simulate **Game**$_0$. Otherwise, the random will be added to $k_0$ and this will simulate **Game**$_1$.

**Guess** Finally, $\mathcal{A}$ outputs a bit $b'$. $\mathcal{B}$ sends $b'$ to the challenger.

□

**Lemma 2.** *Suppose there exists a PPT algorithm $\mathcal{A}$, which can distinguish between **Game**$_1$ and **Game**$_2$ which are indistinguishable with non-negligible advantage $\epsilon$. Then, one can build an algorithm $\mathcal{B}$ which breaks DB-DHES assumption using $\mathcal{A}$ with $\epsilon$.*

Proof. This proof is almost identical as the proof of Lemma 1. The only difference is that the algorithm $\mathcal{B}$ sets $k_1$ using $T$ and sets $k_0$ to be a random from $\mathbb{G}_T$. □

# 7 IMPLEMENTATION

## 7.1 Our System

We implement our scheme using Pairing Based Cryptographic (PBC) Library [17]. We use type A pairing constructed on a supersingular curve [17] to feature our scheme. It requires 512 bits for $\mathbb{G}$ and $\mathbb{G}_T$. Our implementation consists of three parties, a cloud server (CS), an edge server (ES) and the end device (ED). We use Microsoft Azure for CS and a virtual machine running on a general PC for

**Table 2: Specifications**

| Cloud Server (CS) | |
|---|---|
| OS | Ubuntu 18.04.1 LTS |
| CPU | Intel(R) Xeon(R) E5-2673v4@2.30GHz |
| Memory | 8 GB |
| Power | Constant |

| Edge Server (ES) | |
|---|---|
| Guest OS | Ubuntu 16.04 LTS (4 CPUs/8GB RAM) |
| Main OS | Windows 10 (64bits) |
| CPU | Intel(R) Core(TM) i5-7440HQ@2.80GHz |
| Memory | 16 GB |
| Power | Constant |

| End Device (ED) | |
|---|---|
| OS | Raspbian |
| CPU | Broadcom A53(ARMv8)@1.4GHz |
| Memory | 1 GB |
| Power | Constant or Battery |

**Table 3: Communication overhead on ED**

| Schemes | Communication Overhead |
|---|---|
| $GW_{semi}$ | $3\mathbb{G} + \mathbb{G}_T + 2$ S |
| $GW_{adapt}$ | $6\mathbb{G} + 2\,\mathbb{G}_T + \ell$ bits + 3 S + 2*SymCT* |
| $JWMS_{semi}$ | $2\mathbb{G}$ + S |
| $JWMS_{adapt}$ | $4\mathbb{G} + \ell$ bits + S + 2*SymCT* |
| $Ours_{semi}$ | $\mathbb{G} + \mathbb{G}_T$ |
| $Ours_{adapt}$ | $2\mathbb{G} + 2\mathbb{G}_T$ + 2*SymCT* |

ES, Raspberry pi 3B+ for ED. Table 2 shows more details on our implemented system.

## 7.2 Communication Overhead

Our system reduces the decryption overhead in the end device. One of the typical overheads is a communication overhead. The communication, which is the packet transmission between an edge and an end device, consumes the battery power. Reducing communication between them prolongs the battery life of the end device and consequently reduces the maintenance. We define the communication overhead as the amount of total information required for the decryption. In an IBBE scheme such as GW [7] and JWMJ [12], the decryption party usually needs to know all recipients' identities for the decryption. Therefore, even though the size of the ciphertext is constant, the actual amount of information, which has to be transmitted to the decryption party (i.e., an end device), increases linearly with the number of recipients.

Table 3 shows the comparison of the ciphertext which is required to be sent from ES to ED. Both of our schemes require only constant information to be transmitted between ES and ED since there is no parameter increasing on the number of recipients and the ED even does not need to know the identity information of other recipients.

**Figure 2: Decryption overhead in ED**

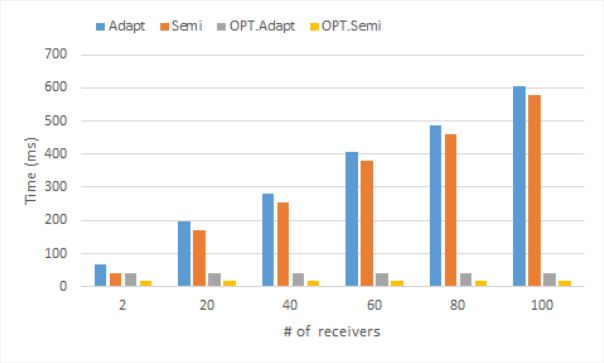
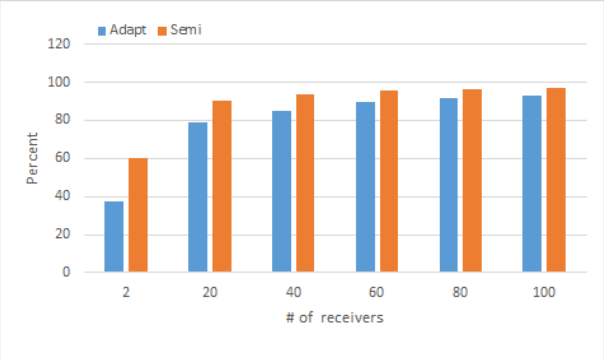



**Figure 3: Reduced decryption overhead in ED (by %)**




It should be noted that the adaptive scheme must include a symmetric ciphertext which is denoted using *"SymCT"* in all compared schemes.

## 7.3 Computation Overhead

Our system aims to minimize the computation overhead on the end devices, which is usually a resource-constrained device such as an IoT. We achieve constant and small end user computation overhead regardless of the number of recipients in both semi-static and adaptively secure schemes. It also results in reducing the overall delay caused by the decryption in the system.

Compared to the JWMJ's IBBE algorithm, which is the most efficient scheme to the best our knowledge, the decryption overhead for the decryption on the end device is reduced by up to 97 per cent (from 579 ms to 16 ms) in the semi-static scheme with 100 receivers. It can be reduced more as the size of the system grows - i.e., the number of the recipients becomes larger.

Figures 2 and 3 show the computation overheads of the end device (ED). The computation overheads in the vertical axis of Figure 2 are computed by measuring the time taken for the decryption in ED. We also draw Figure 3 that shows the reduction using percentages between schemes. The following formula calculates those percentages:

**Figure 4: Overall decryption delay**

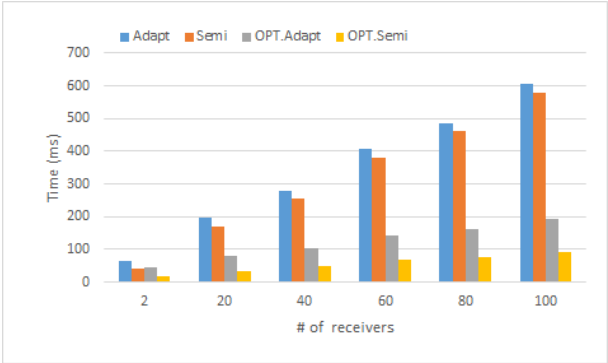
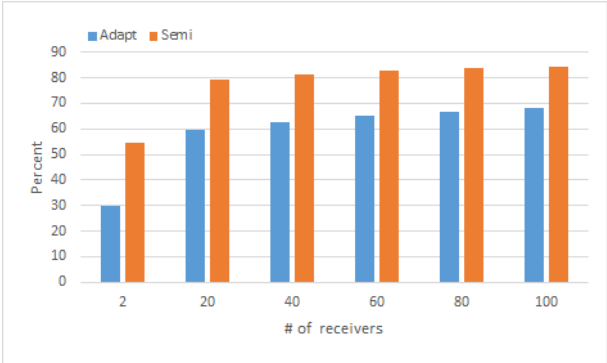



**Figure 5: Reduced overall decryption time (by %)**




$$\frac{\text{IBBE decryption on ED - IBBE-OPT decryption on ED}}{\text{IBBE decryption on ED}} \times 100$$

The saving from the outsourced partial decryption grows as the number of receivers, the horizontal axis, grows because "IBBE decryption on ED - IBBE-OPT decryption on ED" is close to the time to run **PaDec** on ED, which is delegated to ES in IBBE-OPT. Since the computation overheads of **PaDec** increase linearly on the number of receivers and the **CoDec** requires a constant overhead and small, our schemes efficiently reduce the overhead on ED. It is worth noting that the adaptive scheme is slightly less efficient than the semi-static scheme since ED has to perform one symmetric key decryption, which is AES algorithm in our implementation ‡

Our outsourced partial decryption technique not only reduces the computation overhead of ED, but also it reduces the entire delay caused by the decryption in the system. Figures 4 and 5 show the overall time delay caused by the decryption. The figure shows that even in a large number of receivers, such as 100 receivers, the decryption can be done within 100 ms in the semi-static scheme and 200 ms in the adaptive scheme. Among those time, the computation time that required in ED were 41 ms for the adaptive scheme and 16 ms for the semi-static time as shown in Figure 2. Figure 5 depicts the reduced delay by a percentage using the formula as follows:

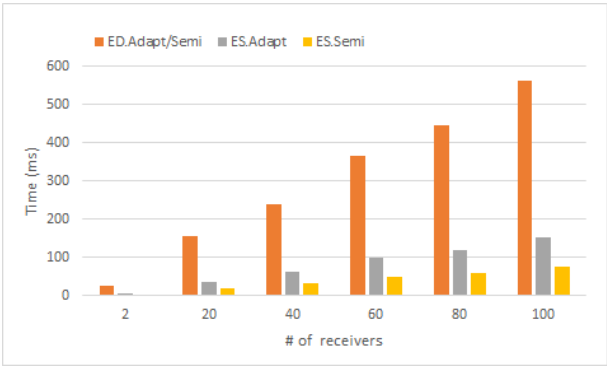
$$\frac{\text{IBBE overall decryption - IBBE-OPT overll decryption}}{\text{IBBE overall decryption}} \times 100$$

The overall delay is reduced because computing **PaDec** in ES is 7 to 10 times faster than computing it in ED as shown in Figure 6. Therefore, the delay is reduced up to 84 percent. In our adaptive scheme, the reduced delay is slightly smaller because **PaDec** has to be performed twice - the edge does not know the bit $u_{ID}$ of user's private key. However, due to the significant difference in the computation power, our scheme still reduced the delay by up to 67 percent.

Figure 6 shows that the computation time of **PaDec** in our system. It should be noted that in a normal IBBE, **PaDec** is performed with the knowledge of $u_{ID}$ because both **PaDec** and **CoDec** are computed in the ED. Therefore, **PaDec** is computed only once and

‡AES encryption/decryption is implemented using OpenSSL.

**Figure 6: Execution times of PaDec**



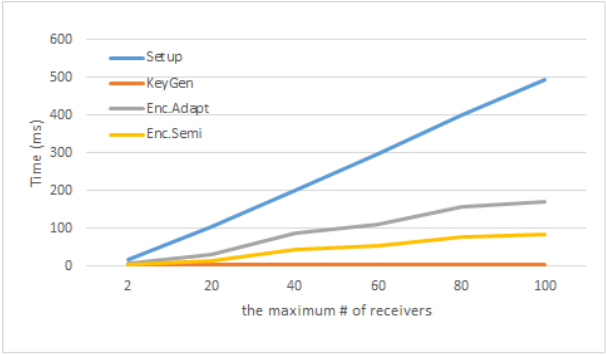

the execution times of semi-static and adaptive schemes are identical.

We also measure the other important execution times performed in the cloud server (CS), which are **Setup**, **Keygen** and **Enc** as depicted in Figure 7. **Setup** algorithm is performed once when the system is set up. Its execution time increases linearly on the maximum number of receivers as shown in Figure 7. **Keygen** is performed independent from the maximum number of receivers. It issues a new key to an ED and takes only 6.4 ms to do so. **Enc** is another key metric that shows the performance of the system. We set the number of receivers for **Enc** to the maximum in our experiment (i.e. it equals to the maximum number of receivers). **Enc** is reasonably fast in our system. It takes about 100ms in the semi-static scheme and about 200ms in the adaptive scheme even the size of receivers is large, 100 receivers.

## 8 CONCLUSION

In this paper, we proposed a new identity-based broadcast encryption with outsourced partial decryption technique to support data encryption in edge computing. Our new technique is based on identity-based broadcast encryption (IBBE), but it significantly reduces the computation and communication overheads required to end devices. We achieve so by delegating the partial decryption to the edge. In particular, our schemes provide the constant-sized transmission between an edge and an end device and the light-weight

**Figure 7: Execution times on CS**




decryption in the end device, which requires only one pairing operation. In edge computing, which consists of resource-constrained IoT devices, this prolongs the battery life of the device and reduces the management burden of the system. In addition to the simple outsourced decryption, our suggested technique provides additional security enhancement. We showed that the ciphertext after the partial decryption can be decrypted by the only one who requests the data using the the security model of IBE. It enables edge to send the encrypted data without requiring an additional authentication. Moreover, we implemented our system in an edge computing system and estimated the efficiency gain that our schemes can achieve. It showed that our system reduces the computation burden on an end device up to 97 per cent and also reduce the latency caused by decryption up to 84 per cent.

## ACKNOWLEDGEMENT

Jongkil Kim, Willy Susilo and Joonsang Baek are partially funded by NSW Cyber Security Network Pilot Grants.

## REFERENCES

[1] Dan Boneh, Craig Gentry, and Brent Waters. 2005. Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys. In *CRYPTO (Lecture Notes in Computer Science)*, Victor Shoup (Ed.), Vol. 3621. Springer, 258–275.

[2] Flavio Bonomi, Rodolfo A. Milito, Preethi Natarajan, and Jiang Zhu. 2014. Fog Computing: A Platform for Internet of Things and Analytics. In *Big Data and Internet of Things: A Roadmap for Smart Environments*. Vol. 546. Springer, 169–186.

[3] Flavio Bonomi, Rodolfo A. Milito, Jiang Zhu, and Sateesh Addepalli. 2012. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing, MCC@SIGCOMM 2012, Helsinki, Finland, August 17, 2012*, Mario Gerla and Dijiang Huang (Eds.). ACM, 13–16.

[4] Cécile Delerablée. 2007. Identity-Based Broadcast Encryption with Constant Size Ciphertexts and Private Keys. In *ASIACRYPT (Lecture Notes in Computer Science)*, Kaoru Kurosawa (Ed.), Vol. 4833. Springer, 200–215.

[5] Yevgeniy Dodis and Nelly Fazio. 2002. Public Key Broadcast Encryption for Stateless Receivers. In *Digital Rights Management Workshop (Lecture Notes in Computer Science)*, Joan Feigenbaum (Ed.), Vol. 2696. Springer, 61–80.

[6] Amos Fiat and Moni Naor. 1993. Broadcast Encryption. In *CRYPTO (Lecture Notes in Computer Science)*, Douglas R. Stinson (Ed.), Vol. 773. Springer, 480–491.

[7] Craig Gentry and Brent Waters. 2009. Adaptive Security in Broadcast Encryption Systems (with Short Ciphertexts). In *EUROCRYPT (Lecture Notes in Computer Science)*, Antoine Joux (Ed.), Vol. 5479. Springer, 171–188.

[8] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. 2006. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM Conference on Computer and Communications Security*, Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati (Eds.). ACM, 89–98.

[9] Matthew Green, Susan Hohenberger, and Brent Waters. 2011. Outsourcing the Decryption of ABE Ciphertexts. In *20th USENIX Security Symposium, San Francisco, CA, USA, August 8-12, 2011, Proceedings*. USENIX Association.

[10] Changhee Hahn, Hyunsoo Kwon, and Junbeom Hur. 2018. Toward Trustworthy Delegation: Verifiable Outsourced Decryption with Tamper-Resistance in Public Cloud Storage. In *11th IEEE International Conference on Cloud Computing, CLOUD 2018, San Francisco, CA, USA, July 2-7, 2018*. IEEE Computer Society, 920–923.

[11] Jonathan Katz and Nan Wang. 2003. Efficiency improvements for signature schemes with tight security reductions. In *ACM Conference on Computer and Communications Security*, Sushil Jajodia, Vijayalakshmi Atluri, and Trent Jaeger (Eds.). ACM, 155–164.

[12] Jongkil Kim, Willy Susilo, Man Ho Au, and Jennifer Seberry. 2013. Efficient Semi-static Secure Broadcast Encryption Scheme. In *Pairing-Based Cryptography - Pairing 2013 - 6th International Conference, Beijing, China, November 22-24, 2013, Revised Selected Papers (LNCS)*, Zhenfu Cao and Fangguo Zhang (Eds.), Vol. 8365. Springer, 62–76.

[13] Junzuo Lai, Robert H. Deng, Chaowen Guan, and Jian Weng. 2013. Attribute-Based Encryption With Verifiable Outsourced Decryption. *IEEE Trans. Information Forensics and Security* 8, 8 (2013), 1343–1354. https://doi.org/10.1109/TIFS.2013.2271848

[14] Jianchang Lai, Yi Mu, Fuchun Guo, Willy Susilo, and Rongmao Chen. 2016. Anonymous Identity-Based Broadcast Encryption with Revocation for File Sharing. In *Information Security and Privacy - 21st Australasian Conference, ACISP 2016, Melbourne, VIC, Australia, July 4-6, 2016, Proceedings, Part II (Lecture Notes in Computer Science)*, Vol. 9723. Springer, 223–239.

[15] J. Li, Y. Wang, Y. Zhang, and J. Han. 2018. Full Verifiability for Outsourced Decryption in Attribute Based Encryption. *IEEE Transactions on Services Computing* (2018), 1–1.

[16] Benoît Libert, Kenneth G. Paterson, and Elizabeth A. Quaglia. 2012. Anonymous Broadcast Encryption: Adaptive Security and Efficient Constructions in the Standard Model. In *Public Key Cryptography - PKC 2012 - 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21-23, 2012. Proceedings (Lecture Notes in Computer Science)*, Vol. 7293. Springer, 206–224.

[17] Ben Lynn. 2007. *On the implementation of pairing-based cryptosystems*. Ph.D. Dissertation. PhD thesis, Stanford Univeristy.

[18] Dalit Naor, Moni Naor, and Jeffery Lotspiech. 2001. Revocation and Tracing Schemes for Stateless Receivers. In *CRYPTO (Lecture Notes in Computer Science)*, Joe Kilian (Ed.), Vol. 2139. Springer, 41–62.

[19] Baodong Qin, Robert H. Deng, Shengli Liu, and Siqi Ma. 2015. Attribute-Based Encryption With Efficient Verifiable Outsourced Decryption. *IEEE Trans. Information Forensics and Security* 10, 7 (2015), 1384–1393. https://doi.org/10.1109/TIFS.2015.2410137

[20] Amit Sahai and Brent Waters. 2005. Fuzzy Identity-Based Encryption. In *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings (Lecture Notes in Computer Science)*, Ronald Cramer (Ed.), Vol. 3494. Springer, 457–473.

[21] Ryuichi Sakai and Jun Furukawa. 2007. Identity-Based Broadcast Encryption. *IACR Cryptology ePrint Archive* 2007 (2007), 217.

[22] Adi Shamir. 1984. Identity-Based Cryptosystems and Signature Schemes. In *CRYPTO (Lecture Notes in Computer Science)*, G. R. Blakley and David Chaum (Eds.), Vol. 196. Springer, 47–53.

[23] Daisuke Wakabayashi. 2014. Tim Cook Says Apple to Add Security Alerts for iCloud Users. *The Wall Street Journal* 5 (2014).

[24] Suzhen Wu, Kuan-Ching Li, Bo Mao, and Minghong Liao. 2017. DAC: Improving storage availability with Deduplication-Assisted Cloud-of-Clouds. *Future Generation Computer Systems* 74 (2017), 190 – 198. https://doi.org/10.1016/j.future.2016.02.001

[25] Peng Xu, Jingnan Li, Wei Wang, and Hai Jin. 2016. Anonymous Identity-Based Broadcast Encryption with Constant Decryption Complexity and Strong Security. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2016, Xi'an, China, May 30 - June 3, 2016*, Xiaofeng Chen, XiaoFeng Wang, and Xinyi Huang (Eds.). ACM, 223–233.

[26] Marcelo Yannuzzi, Rodolfo A. Milito, René Serral-Gracià, D. Montero, and Mario Nemirovsky. 2014. Key ingredients in an IoT recipe: Fog Computing, Cloud computing, and more Fog Computing. In *19th IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, CAMAD 2014, Athens, Greece, December 1-3, 2014*. IEEE, 325–329. http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7016297

[27] Cong Zuo, Jun Shao, Guiyi Wei, Mande Xie, and Min Ji. 2018. CCA-secure ABE with outsourced decryption for fog computing. *Future Generation Computer Systems* 78 (2018), 730 – 738. https://doi.org/10.1016/j.future.2016.10.028