How to Construct Interval Encryption from Binary Tree Encryption

Huang Lin*, Zhenfu Cao, Xiaohui Liang, Muxin Zhou,
Haojin Zhu, and Dongsheng Xing

Department of Computer Science and Engineering, Shanghai Jiao Tong University
faustlin@sjtu.edu.cn*

Abstract. In a broadcast encryption system with a total of n users, each user is assigned with a unique index $i \in [1, n]$. An encryptor can choose a receiver set $S \subseteq [1, n]$ freely and encrypt a message for the recipients in S such that only those receivers can open the message. The transmission overload of most previous broadcast encryption systems grows in line with the number of revoked users r and thus they are suitable for the scenario where the target receiver set is large when $r \ll n$ holds. Some other recently proposed constructions for arbitrary receiver set require a unreasonably large user storage and long decryption time. On the other hand, it is observed that, in a practical broadcast encryption system, the receiver set can be regarded as a collection of k natural intervals, where the interval number k should be much less than r for most cases. This observation motivates us to introduce a novel type of encryption, called interval encryption, which could realize a more efficient broadcast encryption. To achieve this, we first present a generic way to transform a binary tree encryption scheme into interval encryption. One concrete instantiation of this method based on the hierarchical identity based encryption scheme by Boneh et al. only requires a $O(k)$ transmission cost and $O(\log n)$ private storage consumption, while the decryption is dominated by $O(\log n)$ group operations. With detailed performance analysis, we demonstrate that the proposed interval encryption strategy has the superiority on improved efficiency and thus is expected to serve as a more efficient solution in more cases than the traditional systems in practice. Interestingly, our methodology can also be employed to transform a fully secure hierarchical identity based encryption scheme proposed by Lewko and Waters into an adaptively secure interval encryption scheme with a $O(k)$ transmission cost and $O(\log n)$ private storage consumption. Finally, we also discuss several other promising applications of interval encryption.

Keywords: Interval encryption, Public key broadcast encryption, Binary tree encryption, Hierarchical IBE.

1 Introduction

A broadcast encryption (BE) scheme enables a broadcaster to choose a subset S of n users, who are listening on the broadcast channel and encrypt a message for this subset. Any user in S is allowed to successfully decrypt the message while even if all the users outside of S collude together they can not obtain any useful information on the

J. Zhou and M. Yung (Eds.): ACNS 2010, LNCS 6123, pp. 19–34, 2010.

broadcast message. In the following, we also use r to represent the number of revoked users, i.e., $r = n - |S|$ where $|S|$ is the size of S. Compared with a private key broadcast encryption scheme [25,1], a public key broadcast encryption has the benefit that there is no need for the users to pre-share any private information. Therefore, in this study, we mainly focus on pubic key broadcast encryption. Three efficiency parameters of a broadcast encryption scheme are of our major concern: the transmission cost, user storage, and the decryption time.

1.1 Related Work

The transmission overload of most current public key broadcast encryption constructions will grow along with increase of the revocation number r. Naor et al. [22] presented a BE construction (NNL method) with an average ciphertext size of $1.38r$ and private key size $O(\log^2 n)$. The private key size is further improved to $O(\log^{1+\epsilon} n), 0 < \epsilon < 1$ in HS construction [18], where the ciphertext size blows up with a $\frac{1}{\epsilon}$ factor. The private key size is further improved to $O(\log n)$ by Goodrich et al. [16]. Dodis and Fazio [12] presented a generic method (DF transformation) to transfer the NNL method and HS construction into a public key broadcast system using hierarchical identity based encryption (HIBE). The transmission overload remains unchanged and the private key consists of $O(\log^2 n)$ and $O(\log^{1+\epsilon} n)$ HIBE node secret keys if DF transformation is instantiated with BBG HIBE [3]. The security is reduced to standard Decisional BDHE assumption and the decryption time cost is $O(\log n)$. The decryption time is then improved to constant by Liu and Teng [21]. However, their security is reduced to decisional BDH assumption in the random oracle model. Recently, Sahai and Waters proposed a broadcast encryption system with a transmission overload linearly dependent on r and constant storage cost. However, the decryption cost is linearly dependent on r and the security is reduced to a complex assumption called q-MEBDH assumption. Actually, it has been pointed out in [19] that at least one key per each revoked user should be included in the transmission overhead and hence r might be the lower bound of the transmission overload in any broadcast encryption scheme with reasonable decryption computational and storage cost. Therefore, constructing a BE system with a transmission overload lower than r as well as reasonable user storage and computational cost is still an open problem, which is one of the major motivation of this paper.

On the other hand, there are two major application scenarios [5] for broadcast encryption: applications where we broadcast to large sets, namely sets of size $n - r$ for $r \ll n$ and applications where we broadcast to small sets, namely sets of size $|S| \ll n$. Apparently, a broadcast encryption system with a transmission overload dependent on r is not efficient when r grows, and especially it fails to be an optimal choice for the second kind of application where r is very close to n. Before BGW proposed their construction [5], the only suitable solution for the latter scenario is the trivial solution, i.e., encrypting the message under each recipient's key.

In order to construct a BE scheme suitable for *arbitrary* receiver sets, we need to break the barrier of r. BGW [5] proposed an elegant BE scheme with constant size ciphertext as the first attempt to solve this problem. Although the ciphertext and private key size of their construction is constant, the public key material is linearly dependent on n. The public key must be accessible to any decryptor, which implies a high storage

cost of size $O(n)$. This makes their system unsuitable for the application scenario where users have only limited storage capability [24]. Their underlying assumption is the standard Decisional BDHE assumption. Later, Delerablee [11] proposed a BE construction where the public key size depends on the maximum size of S while both ciphertext and private key remain constant size. However, this still does not serve as an efficient solution for applications where the receiver set is large, namely $r \ll n$. The security of this construction is reduced to a complex assumption called GDHE assumption in the random oracle model. Besides, the decryption of both constructions is not efficient. The decryption cost of the BGW construction depends on n, and the decryption of Delerablee's construction requires $O(|S|)$ operations.

1.2 Our Contribution

In this paper, we study this problem from a brand-new angle and a more practical point of view. The basic motivation comes from the following observation: in a broadcast encryption system with n users, where each user is assigned with an index $i \in [1, n]$. The receiver set S can be regarded as a collection of k intervals. Considering the fact that the number of intervals containing in S is always less than $r + 1$ and in the best cases k could even be much less than r, the system performance can be dramatically increased if the transmission overhead of the broadcast encryption system is only determined by the interval number k while irrelevant of r. In this study, we will use more detailed performance analysis and simulation to show that a BE construction based on k is always more efficient than the previous scheme dependent on r, and suitable for more cases in practice.

In order to realize a broadcast encryption system with a transmission overload dependent on k, this paper proposes a new type of encryption called interval encryption. In interval encryption, a message is encrypted under a collection of natural intervals $S = \bigcup_{j=1}^{k} NI_j$, where NI_j is a natural interval in $[1, n]$. Each receiver is identified by a unique natural number $i \in [1, n]$ and assigned with the respective private key. The decryption is successful if and only if the natural number i belongs to S.

We present a generic methodology which can transfer a series of binary tree encryption scheme into interval encryption. We illustrate the basic methodology using the BBG HIBE scheme [3]. The construction achieves a ciphertext size of $O(k)$, and $O(\log n)$ private storage. The decryption is dominated by at most $O(\log n)$ group operations. The security is reduced to the Decisional BDHE assumption. We note that one of the best public key BE schemes under this assumption is the DF transformation of the HS construction which requires a transmission overload of $O(r/\epsilon)$ size and the private key consists of $O(\log^{1+\epsilon} n)$ HIBE node secret keys, where $0 < \epsilon < 1$.

We also apply our basic methodology to the fully secure HIBE [20] scheme proposed by Lewko and Waters to present an adaptively secure interval encryption scheme. Gentry and Waters [15] proposed the first adaptively secure broadcast encryption scheme under a complex bilinear assumption. The public parameter size of their construction is of $O(|S|)$. The private key size is constant, and the ciphertext size of their construction is of $O(max|S|)$. After that, Waters [26] gave the first short ciphertext adaptively

secure broadcast encryption system under static (i.e. non q-based) assumptions. However, both of the public parameter and private key size are linearly dependent on n. The public parameter of our construction is of size $O(\log n)$ and the ciphertext size is of $O(k)$. It only requires $O(\log n)$ private storage. In other words, our construction serves as one of the most efficient adaptively secure broadcast encryption systems. Besides, our construction also reduces its security to static assumptions.

Since we consider the proposal of this new concept and the corresponding methodology one of our major contributions, an inclusive extended interval encryption is proposed as another illustration of the power of our basic methodology. A message is encrypted under a collection of intervals $S = \bigcup_{j=1}^{k} NI_j$ in this extended construction. A user's private key corresponds to a certain interval NI_ω. The decryption is successful if and only if there's at least one interval $NI_j, j \in [1, k]$ such that $NI_\omega \subseteq NI_j$. The construction also provides user with delegation capability. We also discuss several interesting applications of interval encryption. In particular, we propose a useful concept of range attribute based encryption and present an efficient construction from interval encryption.

1.3 Organization

At first, some preliminaries will be given in Section 2. As an important step of understanding the primitive idea of our construction, we'll introduce the notion of binary tree encryption and a different view on forward secure encryption constructed from binary tree encryption in Section 3. The notations used in this paper are introduced in Section 4. A generic transformation from binary tree encryption to interval encryption will be presented in Section 5. In Section 6, we'll give our concrete instantiations based on BBG HIBE and then discuss the system performance in details. In section 7, we introduce how to present an inclusive extended interval encryption using our method. How to construct an efficient adaptively secure interval encryption scheme is shown in section 8. At last, some interesting applications and extensions of interval encryption, including how to construct a range attribute based encryption from interval encryption, are given with some open problems in Section 9.

2 Preliminaries

2.1 Assumptions

Bilinear maps [23] are crucial to our construction. A pairing is an efficiently computable, non-degenerate function, $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, with the bilinearity property that $\hat{e}(g^r, g^s) = \hat{e}(g, g)^{rs}$. Here, $\mathbb{G}_1$, and $\mathbb{G}_2$ are all multiplicative groups of prime order p, respectively generated by g and $\hat{e}(g, g)$.

The security proof of our constructions relies on the Decisional $d+1$ BDHE assumption, which can be stated as [8]: Given a tuple $[h, g, g^{\alpha}, g^{(\alpha^2)}, \cdots, g^{\alpha^d}, g^{(\alpha^{d+2})}, \cdots, g^{(\alpha^{2d})}, Z] \in \mathbb{G}_1^{2d+1} \times \mathbb{G}_2$ for a random exponent $\alpha \in \mathbb{Z}_p$, decide whether $Z = \hat{e}(g, h)^{\alpha^{d+1}}$.

2.2 Security Definitions

Our construction is a Key Encapsulation Mechanism (KEM)[1], thus long messages can be encrypted under a short symmetric key. An interval encryption scheme is made up of four randomized algorithms:

Setup(n)**.** Takes as input a natural interval $[1, n]$. It outputs a public key PK and the system master key SK_ε.

PvkGen(ω, SK_ε)**.** Takes as input a natural number $\omega \in [1, n]$ and the system master key SK_ε. It outputs a private key D_ω.

Encrypt(S, PK)**.** Takes as input a public key PK, and a k-wise natural interval set $S = \bigcup_{j=1}^{k} NI_j$ where $NI_j = [l_j, r_j]$ satisfying $1 \le l_1 \le r_1 < l_2 \le r_2 \cdots < l_k \le r_k \le n$. For $j \in [1, k]$, it outputs k pairs $\{\mathrm{Hdr}_j, K_j\}$. We call $\mathrm{Hdr} = \{\mathrm{Hdr}_j\}_{j=1}^{k}$ the header and $K = \{K_j\}_{j=1}^{k}$ the message encryption keys.

Let M be a message that should be decipherable precisely by the receivers holding the private key corresponding to $\omega \in S$. For $j \in [1, k]$, let C_j be the encryption of M under the message encryption key K_j. Let C_M be the collection of these encryption, namely $C_M = \{C_j\}_{j=1}^{k}$. The whole ciphertext consists of (S, Hdr, C_M).

Decrypt $(S, \omega, D_\omega, \mathrm{Hdr}, PK)$ **.** Takes as input a k-wise natural interval set $S = \bigcup_{j=1}^{k} NI_j$ and the private key D_ω for a natural number $\omega \in [1, n]$, a header Hdr, a public key PK. If $\omega \in NI_j, 1 \le j \le k$, then the algorithm outputs the corresponding message encryption key $K_j \in \mathcal{K}$.

We say the system to be correct, if and only if that for all k-wise natural interval sets $S = \bigcup_{j=1}^{k} NI_j$ and natural numbers $\omega \in NI_j$ (where $j \in [1, k]$), if $PK \xleftarrow{R}$ **Setup**(n), $D_\omega \xleftarrow{R}$ **PvkGen**(ω, SK_ε) and $(\mathrm{Hdr}, K) \xleftarrow{R}$ **Encrypt**(S, PK), then **Decrypt**$(S, \omega, D_\omega, \mathrm{Hdr}, PK)$=$K_j$. The concept of interval encryption is close to private linear broadcast encryption (PLBE) mentioned in [7], and can be viewed as an extension of PLBE.

Semantic Security(IND-sI-CPA) **.** The selective interval game is very similar to that of BE [11], and it forms as follow:

Init. The adversary outputs a k-wise natural interval set $S^* = \bigcup_{j=1}^{k} NI_j^*$, where $NI_j^* = [l_j^*, r_j^*]$ satisfying $1 \le l_1^* \le r_1^* < l_2^* \le r_2^* \cdots < l_k^* \le r_k^* \le n$, which it wishes to attack.

Setup. The challenger runs **Setup**(n) to obtain a public key PK for the adversary.

Phase 1. The adversary issues query for private key of $\omega \notin S^*$.

Challenge . The challenger runs algorithm *Encrypt* to obtain $(\mathrm{Hdr}^*, K) \xleftarrow{R}$ *Encrypt* (S^*, PK) where $K \in \mathcal{K}^k$. Next, the challenger picks a random $\beta \in \{0, 1\}$. It sets $K^* = K$ if $\beta = 1$ and sets K^* to a random string of length equal to $|K|$ otherwise. It then sends Hdr^*, K^* to the adversary.

Phase 2. Same as phase 1.

Guess . The adversary outputs its guess $\beta' \in \{0, 1\}$ for β and wins the game if $\beta' = \beta$.

[1] We adopt KEM for the ease of comparison since all the BE constructions in the literature employ the same mechanism.

The adversary's advantage is the absolute value of the difference between its success probability and $\frac{1}{2}$.

Definition 1. *An interval encryption scheme is selective-interval chosen plaintext secure (IND-sI-CPA) if all polynomial time adversaries have at most a negligible advantage in winning the above security game.*

The adaptive CPA security can be defined in a similar way except that there is no **Init** stage in the adaptive game and the challenge interval S^* in the **Challenge** stage should be provided under the restriction that none of the identities ω for the key queries of **Phase 1** and **Phase 2** belongs to S^*, i.e., $\omega \notin S^*$.

The ultimate security goal is to realize IND-CCA security where the adversary doesn't need to choose the interval set at the beginning and is provided with a decryption oracle. However, this paper concentrates on IND-sI-CPA security, and leaves the formal definition of IND-CCA security in the full version.

3 Binary Tree Encryption and a Different View on Forward Secure Encryption

The concept of binary tree encryption (BTE) was first proposed by Canetti, et al [10]. BTE is a relaxation of hierarchical identity-based encryption (HIBE) [14]. As in HIBE, a "master" public key PK is associated with a binary tree in BTE; each node ω in this tree has a corresponding secret key SK_ω. To encrypt a message "targeted" for some node, one uses both PK and the name of the target node; the resulting ciphertext can then be decrypted using the secret key of the target node. Moreover, as in HIBE the secret key of any node can be employed to derive the secret keys for the children of that node. The only difference between HIBE and BTE is that the latter insists on a *binary* tree, where each non-leaf node only has two child nodes.

Technically speaking, forward secure encryption (FSE) is an elegant application of BTE. Let the depth of a binary tree be d which implies it has $n = 2^d$ leaf nodes. In a FSE scheme, the lifetime of a system is divided into $n = 2^d$ time periods, each of which is associated with a unique leaf node of the tree. A user holding a private key for time period ω can open all the messages encrypted under the subsequent time periods, namely $\omega' \in [\omega, n]$. The private key D_ω in a FSE construction contains the node secret keys SK_ω for the leaf node ω as well as node secret keys for the right siblings of the nodes on the path from the root to node ω, where all these node secret keys come from the underlying BTE scheme. To encrypt a message for a certain period ω', one uses both PK and the name of respective leaf node ω' as in the BTE scheme; the resulting ciphertext can then be decrypted using node secret key $SK_{\omega'}$, which is also similar to the BTE scheme. As shown in Fig. 1. a, a private key D_2 containing the node secret keys SK_2, SK_c, SK_b can be used to derive all the node secret keys for leaf nodes falling into the interval $[2, 8]$. Therefore, D_2 can be used to open all the messages encrypted under time periods in the interval $[2, 8]$.

Indeed, forward secure encryption can be viewed as a special case of interval encryption. As shown in Fig. 1. a, if we use ciphertext C_4 encrypted under leaf node 4 to represent the interval $[1, 4]$, then only the private key for time period $\omega \in [1, 4]$ can

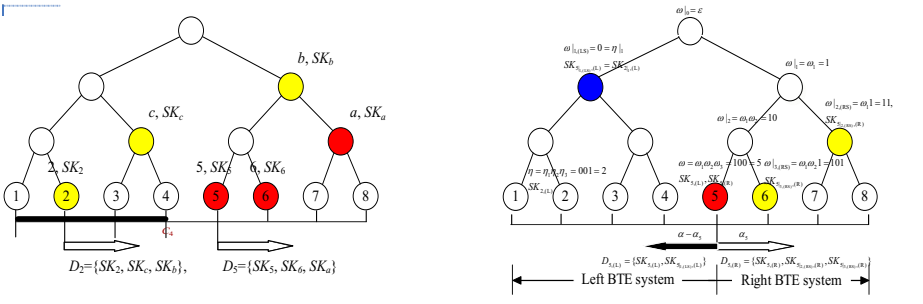

Fig. 1. (a). The key distribution mode of forward secure encryption, C_4 represents the interval $[1,4]$, and the private key for a user ω can be used to derive the node secret keys for all the nodes in the interval $[\omega, n]$ (b). Key distribution mode of interval encryption: we let $\omega = 5$ here. The respective private key D_5 contains left private key$D_{5,(\mathrm{L})}$ and right private key $D_{5,(\mathrm{R})}$. $D_{5,(\mathrm{L})} = \{SK_{5,(\mathrm{L})}, SK_{5|_{1,(\mathrm{LS}),(\mathrm{L})}}\}$ which are derived from left master key $\alpha - \alpha_5$. Similarly, we have $D_{5,(\mathrm{R})} = \{SK_{5,(\mathrm{R})}, SK_{5|_{2,(\mathrm{RS}),(\mathrm{R})}}, SK_{5|_{3,(\mathrm{RS}),(\mathrm{R})}}\}$ derived from right master key α_5. Let the left bound η of an interval be 2 here, then $SK_{2,(\mathrm{L})}$ can be derived from $SK_{2|_{1,(\mathrm{L})}}$ which is equal to $SK_{5|_{1,(\mathrm{LS}),(\mathrm{L})}}$ belonging to $D_{5,(\mathrm{L})}$.

be used to open the message, e.g., D_2 could be used for the decryption of C_4 because SK_4 can be derived from SK_c, which belongs to D_2. However, D_5 cannot be used for decrypting C_4 for it is impossible to deduce SK_4 from any node secret keys included in D_5.

In the remainder of this paper, we use a right direction arrow from a certain leaf node (or the corresponding index in the axis) to denote this particular private key distribution mode. A right direction arrow from a leaf node ω means that all the node secret keys of the leaf nodes in the interval $[\omega, n]$ are computable from its own private key. Therefore, this private key can be used to open all the message encrypted under these nodes. Besides, we also use a left direction arrow from a leaf node ω to denote an opposite decryption ability, namely the respective private key can be used to open all the messages encrypted under the leaf nodes in the interval $[1, \omega]$. It is feasible by simply assigning a user with the node secret keys for node ω as well as node secret keys for the left siblings of all the nodes on the path from the root to ω. Generally speaking, a FSE construction is treated as a special interval encryption scheme in which the encryptor can set the interval form as $[1, j]$. The upper bound j depends on which leaf node the ciphertext corresponds to. Now, our goal is to realize an interval encryption scheme covering multiple intervals, each of which has two freely chosen bounds determined by the encryptor.

4 Notation

We inherit most notations from the underlying BTE and FSE [9] construction. Recall that d denotes the depth of the tree, and $n = 2^d$ is the number of leaf nodes. We set the root node to be ε by convention. The other nodes on the tree have an associated name chosen from $\{0,1\}^{\leq d}$. The left child of a node is concatenated with 0, and the right child

is concatenated with 1. Therefore each leaf node will also have an associated binary name $[\omega_1\omega_2\cdots\omega_d]$. We also let a natural number $\omega \in [1, n]$ associate with the ω-th leaf node of the binary tree (starting from left to right). We implicitly let $\omega = [\omega_1\omega_2\cdots\omega_d]$ in the remainder of this paper. The j-bit prefix of a string $\omega = [\omega_1\omega_2\cdots\omega_d]$ is denoted by $\omega|_j$, namely $\omega|_j$=$[\omega_1\omega_2\cdots\omega_j]$. We implicitly set $\omega|_0 = \varepsilon$ and $\omega|_d = \omega$. It is easy to observe that a set of nodes $\omega|_j$, $j \in [1, d]$ corresponds to the nodes on the path from the root to the leaf node ω (see Fig. 1. (b)). Besides, we use $\omega|_{j,(\mathrm{RS})}$ or $\omega|_{j,(\mathrm{LS})}$ to denote the right or left sibling of $\omega|_j$ respectively if $\omega|_j$ has such a sibling. Namely, $\omega|_{j,(\mathrm{RS})} = [\omega_1\omega_2\cdots\omega_{j-1}1]$ or $\omega|_{j,(\mathrm{LS})} = [\omega_1\omega_2\cdots\omega_{j-1}0]$.

Generally speaking, our BE system consists of two parallel BTE systems: the right BTE system and the left BTE system. The right BTE system covers all the leaf nodes in the interval $[\omega, n]$ and the left BTE system covers all the leaf nodes in the interval $[1, \omega]$. User ω will be assigned with a unique right master key and a left master key. All the node secret keys or private keys for ω in the right BTE system are derived from the right master key while its node secret keys or private keys in the left BTE system are derived from the left master key. We use two different subindexes (L) or (R) in the notations of all these keys to distinguish the left or right BTE system they correspond to respectively.

5 Primitive Idea: A Generic Transformation from BTE to Interval Encryption

5.1 Trivial Constructions

A trivial interval encryption scheme can be given directly from attribute based encryption [17] if one treats $\log n$ bits to represent a number from 1 to n as attributes and builds an access tree allowing specific intervals. However, even the most efficient trivial methodology would inevitably result in an interval encryption construction with a ciphertext size of $O(k \log n)$, where k is the number of intervals. As introduced in the introduction, our goal is to realize a broadcast encryption system in which the ciphertext size is determined by the number of intervals k. *If all the messages are only encrypted under the bounds of each interval like in the FSE scheme*, then this goal is reachable. However, how to make sure that only those receivers with an index within two bounds of each interval can open the message still represents a challenge.

5.2 A Generic Transformation from BTE to Interval Encryption

Yet there remains some difficulties to conquer. The first difficulty is how to differentiate the decryption ability of an index in and outside of an interval. Taking the interval [3, 6] shown in Fig. 2. a for instance, we could easily find the required difference if we project two oppositely direction arrows from each index in the axis, where the connotation of the arrows can be found in our exposition of the last paragraph in Sec. 3. The key observation to our transformation is that: *the two opposite direction arrows starting from index* 5 *can cross both bounds* 3 *and* 6 *respectively and therefore decrypt the corresponding partial ciphertext in two different manners* (We will show how

to differentiate the partial decryption from two different directions, and how this will eventually lead to successful generation of the corresponding message encryption key in the sequel). *However, only one unique direction arrow from index* 2 *or* 7 *can cross the two bounds, i.e., only the right direction arrow from index* 2 *can cross* 3 *and* 6 *while only the left direction arrow from index* 7 *can cross* 3 *and* 6. *This implies that those outside of an interval can only decrypt the partial ciphertext in a unique manner*. The private key for right direction arrow is called right private key in the concrete construction while the one for left direction arrow is left private key.

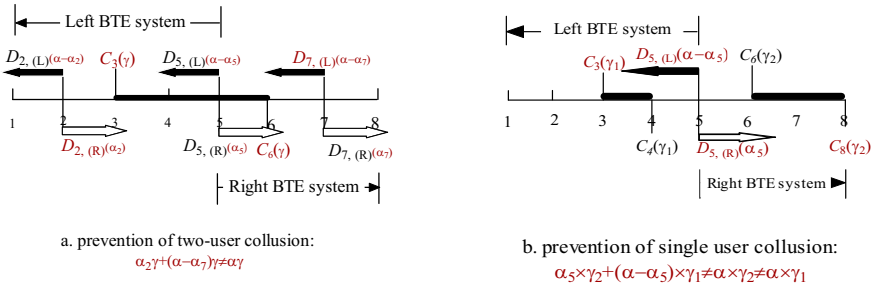

Fig. 2. Collusion and its prevention

We require that the master key of the underlying BTE or HIBE scheme only contains *one group element*. The message encryption key for each interval corresponds to $\alpha \cdot \gamma$ in the exponent of a pairing, where α is the system master key and γ is a randomness chosen by the encryptor. For each user ω, we choose a random number α_ω and split the master key α into two parts: one is the right master key α_ω, which serves as the root master key for the right BTE system, from which the right private key $D_{\omega,(\mathrm{R})}$ of ω is derived; the other part is the left master key $\alpha - \alpha_\omega$, i.e., the root master key for the left BTE system, from which the left private key $D_{\omega,(\mathrm{L})}$ of ω is derived. It is observable that the two private keys for ω can be distributed similarly to a FSE scheme as shown in Fig. 1. b. Consequently, a partial decryption using the user's right private key contains $\alpha_\omega \cdot \gamma$ in the exponent of a pairing while a partial decryption using the left private key will have $(\alpha - \alpha_\omega) \cdot \gamma$ in its exponent. Then, the message encryption key containing $\alpha \cdot \gamma$ in its exponent will be recovered since $\alpha = \alpha_\omega + \alpha - \alpha_\omega$ holds.

In this way, we can actually prevent a possible collusion attack called two-user collusion. For example (shown in Fig. 2. a), a user $\omega = 7$ with a left private key $D_{7,(\mathrm{L})}$ (which could decrypt the partial ciphertext C_3) and a user $\omega = 2$ with right private key $D_{2,(\mathrm{R})}$ (which could decrypt the partial ciphertext C_6) might collude to open the message aiming for interval $[3, 6]$ (since they could also complete the partial decryption in two different manners) although neither of them is in this particular interval. In our system, the partial decryption from $D_{7,(\mathrm{L})}$ will contain $(\alpha - \alpha_7) \cdot \gamma$ while the partial decryption from $D_{2,(\mathrm{R})}$ contains $\alpha_2 \cdot \gamma$ in their exponents, and hence the collusion will fail since there's no way for them to incorporate $\alpha \cdot \gamma$ in the final step.

Besides, we require the encryptor to use a unique randomness γ_j while generating the ciphertext for each interval NI_j. This aims to prevent another attack called a single-user

collusion. This attack only occurs in the scenario with multiple intervals (where $k \geq 1$). For instance (shown in Fig. 2. b), in an interval encryption system with two intervals $[3, 4] \bigcup [6, 8]$, the partial decryption on C_3 from the left private key $D_{5,(\mathrm{L})}$ contains $\alpha - \alpha_5$ and the partial decryption on C_8 from the right private key $D_{5,(\mathrm{R})}$ contains the other half randomness α_5, the message encryption key corresponding to $\alpha \cdot \gamma$ might be recovered if these two intervals use the same randomness. However, a unique randomness for each interval can guarantee that only a user within a certain interval can successfully open the message. For example (as shown in Fig. 2. b), a randomness γ_1 is used in the ciphertext for interval $[C_3, C_4]$ and γ_2 is used in the encryption for interval $[C_6, C_8]$. The message encryption keys of these two intervals correspond to $\alpha \cdot \gamma_1$ and $\alpha \cdot \gamma_2$ respectively. A single-user collusion fails since the randomized partial decryption $(\alpha - \alpha_5)\gamma_1$ and $\alpha_5\gamma_2$ won't incorporate a meaningful encryption key in the final step (see Fig. 2. b).

Although the proposed methodology is generic, it is not fully generic since it somehow relies on the property of bilinear mapping. Therefore, we only illustrate our methodology using concrete examples rather than providing a formal description of a generic interval encryption system in the following sections.

6 Basic Construction: A Concrete Instantiation Based on BBG HIBE

In the following section, we'll describe how the proposed methodology can be applied to the BBG HIBE(viewed as a binary tree encryption scheme here) construction [3] to propose an interval encryption system. Note that there is an additional algorithm **DeckeyDer** $(D_\omega = \{D_{\omega,(\mathrm{L})}, D_{\omega,(\mathrm{R})}\}, \zeta, \eta)$ compared with the original definition of interval encryption in Sec. 2.2. This algorithm is a preliminary step for the decryption algorithm, and we treat it as an independent algorithm for clarity. Besides, there's an additional slightly technical modification to the underlying BTE construction in the sense that we basically have two concrete instantiations of a hash function to guarantee that we could cover both the two bounds of each interval in the security proof.

Let $\mathbb{G}_1$ be a bilinear group of prime order p, and let g be a generator of $\mathbb{G}_1$. In addition, let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ denote the bilinear map. A security parameter, κ, will determine the size of the groups. Assume the system accommodates $n = 2^d$ users, where d is an integer.

Setup(n)**:** Select a random $\alpha \in \mathbb{Z}_p$ and set $g_1 = g^\alpha$. Choose random elements g_2, $g_{3,(\mathrm{L})}$, $g_{3,(\mathrm{R})}$, $h_{1,(\mathrm{L})}, \cdots, h_{d,(\mathrm{L})}$, $h_{1,(\mathrm{R})}, \cdots, h_{d,(\mathrm{R})}$ from $\mathbb{G}_1$.

The public key is $PK = (g, g_1, g_2, g_{3,(\mathrm{L})}, g_{3,(\mathrm{R})}, h_{1,(\mathrm{L})}, \cdots, h_{d,(\mathrm{L})}, h_{1,(\mathrm{R})}, \cdots, h_{d,(\mathrm{R})})$. For a binary string $v = [v_1 v_2 \cdots v_j]$ where $j \in [1, d]$, define two publicly computable functions: $F_{(\mathrm{L})}(v) = g_{3,(\mathrm{L})} \cdot \prod_{i=1}^{j} h_{i,(\mathrm{L})}^{v_i}$ and $F_{(\mathrm{R})}(v) = g_{3,(\mathrm{R})} \cdot \prod_{i=1}^{j} h_{i,(\mathrm{R})}^{v_i}$. The system master key is $SK_\varepsilon = g_2^\alpha$.

PvkGen(ω, SK_ε)**:** For receiver $\omega = [\omega_1 \omega_2 \cdots \omega_d]$ which is associated with the ω-th leaf node (starting from left to right), the algorithm first chooses a random number α_ω. The right master key for ω is $SK_{\varepsilon,(\mathrm{R})} = g_2^{\alpha_\omega}$, and the left master key is $SK_{\varepsilon,(\mathrm{L})} = g_2^{\alpha - \alpha_\omega}$. The algorithm first generates two node secret keys $SK_{\omega,(\mathrm{R})} = [g_2^{\alpha_\omega}(F_{(\mathrm{R})}(\omega))^{r_\omega}, g^{r_\omega}]$ and $SK_{\omega,(\mathrm{L})} = [g_2^{\alpha - \alpha_\omega}(F_{(\mathrm{L})}(\omega))^{r_\omega}, g^{r_\omega}]$ for leaf node ω where r_ω is a random number from $\mathbb{Z}_p$.

For all the nodes $\omega|_j$, $j = 1, \cdots, d$ on the path from the root to the leaf node ω, if it has a right sibling $\omega|_{j,(\mathrm{RS})}=[\omega_1\omega_2 \cdots \omega_{j-1}1]$, the algorithm uses the right master key to generate the respective node secret key as $SK_{\omega|_{j,(\mathrm{RS}),(\mathrm{R})}}=[g_2^{\alpha_\omega}\ (F_{(\mathrm{R})}(\omega|_{j,(\mathrm{RS})}))^{r_j},\ g^{r_j},\ h_{j+1,(\mathrm{R})}^{r_j}, \cdots, h_{d,(\mathrm{R})}^{r_j}]$ where r_j is also a random number; Otherwise the algorithm uses the left master key to generate node secret key for its left sibling $\omega|_{j,(\mathrm{LS})}=[\omega_1\omega_2\cdots\omega_{j-1}0]$ as $SK_{\omega|_{j,(\mathrm{LS}),(\mathrm{L})}}=[g_2^{\alpha-\alpha_\omega}\ (F_{(\mathrm{L})}(\omega|_{j,(\mathrm{LS})}))^{r_j},\ g^{r_j},\ h_{j+1,(\mathrm{L})}^{r_j}, \cdots, h_{d,(\mathrm{L})}^{r_j}]$.

Output private key $D_\omega = \{D_{\omega,(\mathrm{R})}, D_{\omega,(\mathrm{L})}\}$, where $D_{\omega,(\mathrm{R})} = \{SK_{\omega,(\mathrm{R})}, SK_{\omega|_{j,(\mathrm{RS}),(\mathrm{R})}}\}_{j\in[1,d]}$ and $D_{\omega,(\mathrm{L})} = \{SK_{\omega,(\mathrm{L})}, SK_{\omega|_{j,(\mathrm{LS}),(\mathrm{L})}}\}_{j\in[1,d]}$.

Encrypt(S, PK)**:** The encryptor first chooses a k-wise natural interval set $S = \bigcup_{j=1}^{k} NI_j$, where $NI_j = [l_j, r_j]$. For each interval, pick γ_j uniformly from $\mathbb{Z}_p$ at random. Let the binary name of the corresponding leaf nodes for the two bounds be $r_j = [r_{j1} \cdots r_{jd}]$ and $l_j = [l_{j1} \cdots l_{jd}]$.

Output the respective ciphertext $C_{l_j}=\{g^{\gamma_j}, (F_{(\mathrm{L})}(l_j))^{\gamma_j}\}$ and $C_{r_j}=\{g^{\gamma_j}, (F_{(\mathrm{R})}(r_j))^{\gamma_j}\}$. Set the message encryption key for each interval NI_j as $K_j=\hat{e}(g_1,\ g_2)^{\gamma_j} \in \mathbb{G}_2$. The collection of these partial ciphertexts constitute the header $\mathrm{Hdr}=\{C_{l_j}, C_{r_j}\}_{j=1}^{k}$.

DeckeyDer $(D_\omega = \{D_{\omega,(\mathrm{L})}, D_{\omega,(\mathrm{R})}\}, \zeta, \eta)$**:** This algorithm derives the node secret key $SK_{\eta,(\mathrm{L})}$ for the lower bound η, and $SK_{\zeta,(\mathrm{R})}$ for the upper bound ζ.

1. Let a natural number $\eta \le \omega$ denote the η-th leaf node, and thus η is on the left of ω in the binary tree. Assume the binary representation of η is $\eta = \eta_1 \cdots \eta_d$. There must exist a node secret key $SK_{\eta|_j,(\mathrm{L})}$, $j \in [1, d]$ which belongs to $D_{\omega,(\mathrm{L})}$ (as shown in Fig. 1. b). Run the derivation algorithm of the underlying BTE scheme iteratively, which means the following steps need to be executed iteratively for $i = j$ to $i = d - 1$:

(a) Let $\eta|_i = \eta_1 \cdots \eta_i$. Parse $SK_{\eta|_i,(\mathrm{L})}$ as $\left(g_2^{\alpha-\alpha_\omega}(F_{(\mathrm{L})}(\eta|_i))^{r_i},\ g^{r_i},\ h_{i+1,(\mathrm{L})}^{r_i}, \cdots,\ h_{d,(\mathrm{L})}^{r_i}\right)=(a_0, a_1, b_{i+1}, \cdots, b_d)$.

(b) Choose random $t \in \mathbb{Z}_p$, and output $SK_{\eta|_{i+1},(\mathrm{L})} = (a_0 \cdot b_{i+1}^{\eta_{i+1}} \cdot (F_{(\mathrm{L})}(\eta|_{i+1}))^t, a_1 \cdot g^t, b_{i+2} \cdot h_{i+2,(\mathrm{L})}^t, \cdots, b_d \cdot h_d^t)$ and set $i = i + 1$.

Finally, it will output a node secret key $SK_{\eta,(\mathrm{L})}=[g_2^{\alpha_\omega}(F_{(\mathrm{L})}(\eta))^{r'},\ g^{r'}]$ for the lower bound η.

2. Let a natural number $\zeta \ge \omega$ denote the ζth leaf node. Assume the binary representation of ζ is $\zeta = \zeta_1 \cdots \zeta_d$. Therefore there must exist a node secret key $SK_{\zeta|_j,(\mathrm{R})}$ which belongs to $D_{\omega,(\mathrm{R})}$. Run the derivation algorithm of the underlying BTE scheme iteratively, which means steps 1(a)-1(b) need to be executed iteratively.

Output a node secret key $SK_{\zeta,(\mathrm{R})}=[g_2^{\alpha_\omega}(F_{(\mathrm{R})}(\zeta))^{r''},\ g^{r''}]$ for the upper bound ζ.

Decrypt $(S, \omega, D_\omega, \mathrm{Hdr}, PK)$**:** If $\omega \in NI_j = [l_j, r_j]$, $1 \le j \le k$ which implies that $l_j \le \omega \le r_j$, then it runs **DeckeyDer** (D_ω, r_j, l_j) to generate decryption key $SK_{r_j,(\mathrm{R})}$ and $SK_{l_j,(\mathrm{L})}$. It obtains the corresponding secret key $SK_{r_j,(\mathrm{R})}=[g_2^{\alpha_\omega}(F_{(\mathrm{R})}(r_j))^{r''},\ g^{r''}]$ and the partial ciphertext for the upper bound $C_{r_j}=\{g^{\gamma_j}, (F_{(\mathrm{R})}(r_j))^{\gamma_j}\}$. Compute $\frac{\hat{e}[g^{\gamma_j},\ g_2^{\alpha_\omega}(F_{(\mathrm{R})}(r_j))^{r''}]}{\hat{e}[g^{r''},\ (F_{(\mathrm{R})}(r_j))^{\gamma_j}]} = \hat{e}(g,\ g_2)^{\gamma_j\alpha_\omega}$. It also obtains the corresponding secret key $SK_{l_j,(\mathrm{L})}=[g_2^{\alpha-\alpha_\omega}(F_{(\mathrm{L})}(l_j))^{r'},\ g^{r'}]$ and the partial ciphertext for the lower bound $C_{l_j}=\{g^{\gamma_j},\ (F_{(\mathrm{L})}(l_j))^{\gamma_j}\}$. Compute $\frac{\hat{e}[g^{\gamma_j},\ g_2^{\alpha-\alpha_\omega}(F_{(\mathrm{L})}(l_j))^{r'}]}{\hat{e}[g^{r'},\ (F_{(\mathrm{L})}(l_j))^{\gamma_j}]} = \hat{e}(g,\ g_2)^{\gamma_j(\alpha-\alpha_\omega)}$. Finally, it manages to compute $\hat{e}(g,\ g_2)^{\gamma_j\alpha_\omega} \cdot \hat{e}(g,\ g_2)^{\gamma_j(\alpha-\alpha_\omega)} = \hat{e}(g^\alpha,\ g_2)^{\gamma_j} = \hat{e}(g_1,\ g_2)^{\gamma_j}$

6.1 Discussion on Efficiency and Security

In this construction, the public key size is $O(\log n)$, and the private key only contains $O(\log n)$ BTE node secret keys. Note that the private key in the DF transformation [12] of the NNL method or the HS construction contains $O(\log^2 n)$ or $O(\log^{1+\epsilon} n)$ node secret keys respectively. It is important to point out that a widely used tool, updateable public storage in the FSE scheme [4], can be also adopted in our proposed interval encryption system to limit the private storage cost to $O(\log n)$. The above efficiency parameters could be further improved if the random oracle is adopted, i.e., the public key size can be reduced to $O(1)$ in this case.

The decryption cost is dominated by the derivation of the two node secret keys. The derivation cost can be reduced to $O(\log n)$ by doing the following computation: in order to deduce the node secret key $SK_{\eta,(\mathrm{L})}=[g_2^{\alpha-\alpha_\omega}\cdot (F_{(\mathrm{L})}(\eta))^{r'}, g^{r'}]=(a'_0, a'_1)$ from $SK_{\eta|_i,(\mathrm{L})}=\left(g_2^{\alpha-\alpha_\omega}(F_{(\mathrm{L})}(\eta|_i))^{r_i}, g^{r_i}, h_{i+1,(\mathrm{L})}^{r_i}, \cdots, h_{d,(\mathrm{L})}^{r_i}\right)=(a_0, a_1, b_{i+1}, \cdots, b_d)$, we could compute $a'_0 = a_0 \cdot \prod_{k=i+1}^{d} b_k^{\eta_k} \cdot (F_{(\mathrm{L})}(\eta))^t, a'_1 = a_1 \cdot g^t$ where we force $r' = r_i + t$. We could deduce the node secret key $SK_{\zeta,(\mathrm{R})}$ from $SK_{\zeta|_i,(\mathrm{R})}$ in a similar way. The overall decryption time is then reduced to $O(\log n)$ since the rest of the decryption procedure only requires a constant number of group operations.

Why is $O(k)$ better: A system with a transmission overhead proportional to k is more efficient than the traditional systems, especially the system where communication load is linearly dependent on r such as the revocation system proposed in [24,27]. To demonstrate that, we compare the performance of both systems in presence of different values for r as well as k. We assume that the total node number n is set to $2^{17} = 131072$ and let r increase from 1 to n. For a specific r, we randomly generate 1000 revoked sets, which correspond to 1000 different interval number k, and thus obtain an average interval number $\overline{k}$ as well as the average transmission overhead of the proposed scheme, which has been shown in Fig. 3. From Fig. 3, it is observed that, when the revocation set is small, the performance of the proposed scheme is very close to the tradition systems, however the difference will be scaled up along with the increase of r. If the revoked set number exceeds 50% of the total number, the communication load of the proposed scheme will decrease with increase of r. It is also observed that the proposed scheme can achieve the best performance in case that r is very large, which further demonstrates that the proposed scheme is suitable for cases when a small receiver set is employed. Compared with the BGW generalized construction [5] with a $\sqrt{n}$ size transmission overload which only serves as a better choice than the trivial solution and the traditional systems when $r > \sqrt{n}$, we have the benefit that our system keeps the advantage of the traditional constructions when r is a small number, namely $r \ll n$.

From the above results, we conclude that the proposed construction fits into more cases than the traditional systems dependent on r, and therefore constitutes a more favorable choice in practice.

The selective security of the proposed construction can be proven secure under the $d + 1$-BDHE assumption, and it's stated as follow. We leave the concrete proof in the full version.

Theorem 1. *If the Decisional $(d + 1)$-BDHE assumption holds in $\mathbb{G}_1, \mathbb{G}_2$, then the proposed interval encryption scheme is selective chosen plaintext secure.*

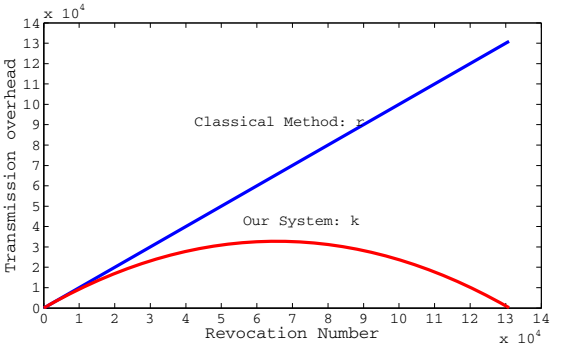

Fig. 3. Comparison between k and r

7 Inclusive Extended Interval Encryption

An inclusive extended interval encryption scheme deals with the scenario where the message is encrypted under a collection of intervals $S = \bigcup_{j=1}^{k} [l_j, r_j]$, and the private key D_ω of a user ω corresponds to an interval $[l_\omega, r_\omega]$. The decryption is successful if and only if there exists at least one interval $[l_j, r_j]$, $j \in [1, k]$ such that $[l_\omega, r_\omega] \subseteq [l_j, r_j]$. To generate a private key corresponding to an interval $[l_\omega, r_\omega]$ (see Fig. 4), we simply generate a left private key $D_{l_\omega,(\mathrm{L})}$ corresponding to the lower bound l_ω using the left master key $\alpha - \alpha_\omega$ as in the basic construction. Similarly we generate a right private key $D_{r_\omega,(\mathrm{R})}$ for the upper bound r_ω using the right master key α_ω. The rest of the above algorithms have no significant differences from those in the basic construction. Furthermore, it is easy to observe that a man holding a private key for an interval $[l_\omega, r_\omega]$ can delegate a private key for another interval $[l_{\omega'}, r_{\omega'}]$ using the **DeckeyDer** algorithm as long as $[l_\omega, r_\omega] \subseteq [l_{\omega'}, r_{\omega'}]$. This is a property somewhat close to a recently proposed concept called inclusive identity based encryption (IBE) [6]. We consider this extended construction of important theoretical interest since there exist very few inclusive constructions [13] since the proposal of inclusive IBE.

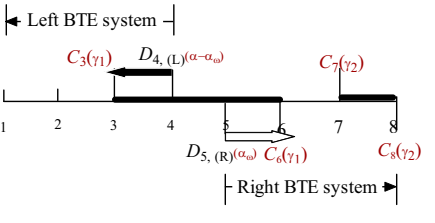

Fig. 4. Extended Interval Encryption: the generation of a private key for an interval [4, 5]

8 Adaptively Secure Interval Encryption

This construction is based on Lewko and Waters' HIBE construction [20]. The basic idea is to apply our proposed transformation method to Lewko and Waters' HIBE scheme, and the concrete construction will be shown in the full version.

9 Extensions and Future Work

9.1 Range Attribute Based Encryption

In a key policy attribute based encryption (ABE) [17], a private key might be associated with an access policy such as "Old man **AND** tall". A man holding this private key can open a message encrypted under an attribute set {"Old man", "tall"} since this attribute set satisfies the above access policy. In practice, the attributes in an attribute set might have certain range and the attributes in an access policy might be assigned with certain concrete evaluations. In the above example, the access policy might be denoted as a formula "Age: 60 **AND** Height: 180 (cm)". A man holding a private key associated with the above policy should be able to open a message encrypted under an attribute set {"Age: 50 to 100", "Height: 175 to 250 (cm)"}. The reason for the successful decryption is that both evaluations of the two attributes fall into the range required in the attribute set and hence the access policy is satisfied. However, A man holding a private key associated with an access policy "Age: 49 **AND** Height: 180 (cm)" cannot decrypt this message since the evaluation "Age: 49" is not within the corresponding range "Age: 50 to 100" in the attribute set. A range ABE scheme is realizable from a traditional ABE scheme. However, the ciphertext will blow up with a $\log n$ factor as shown in our trivial example of constructing interval encryption from ABE.

The proposed interval encryption scheme can be easily modified to a range ABE scheme with a constant ciphertext size. The primitive idea and concrete construction can be found in the full version.

9.2 Interval Encryption under Simpler Assumption

The proposed method also applies to those BTE or HIBE schemes, in which cases their master keys only contain one single group element such as [2,9]. We can construct interval encryption schemes based on the decisional bilinear Diffie-Hellman assumption. The concrete steps are similar to that of Sec.6 and hence trivial. The weakness of these constructions is that the ciphertext size will blow up with a $\log n$ factor compared with the basic construction while the private key size remains $O(\log n)$.

9.3 Encryption under a Graph

Consider the following application: a message might be encrypted under a digital map of a certain territory on earth (which a close two-dimensional graph can represents) and only those who hold a private key for a location in the territory can open the message. This notion might actually intrigue several interesting applications. For example,

a launch order of a certain weapon might be encrypted under a map of a specific region and only those who have a private key corresponding to a location within this region can launch this weapon. Apparently, we could map all the points in a two-dimensional digital map to the points in an one-dimensional axis. We could simply calculate $i = (y-1)c + x$ where (x, y) is a point in a two-dimensional map with width c and height d. If we set $n = c * d$, then all the points can be mapped into an index $i \in [1, n]$. In other words, all the points within the territory of this digital map can be mapped into a collection of intervals. Therefore, the proposed interval encryption provides a solution for the above scenario. The count of intervals depends on the perimeter of this graph.

9.4 Future Work and Open Problems

The reason why the proposed construction can improve the transmission overload relies on the fact that we utilize the difference between the points in and outside a certain interval. How to use the difference between a point in and outside of a graph to reduce the transmission overload, especially to minimize the constant factor k during the multi-dimensional scenario is left as an important open problem. It is possible to borrow some idea from computational geometry to solve this problem. To propose a BTE or HIBE construction with improved efficiency or under a weaker assumption which fits into our framework is very interesting since this directly implies the improvement of interval encryption.

References

1. Attrapadung, N., Imai, H.: Graph-decomposition-based frameworks for subset-cover broadcast encryption and efficient instantiations. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 100–120. Springer, Heidelberg (2005)
2. Boneh, D., Boyen, X.: Efficient selective-ID secure identity-based encryption without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
3. Boneh, D., Boyen, X., Goh, E.-J.: Hierarchical identity based encryption with constant size ciphertext. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (2005)
4. Boneh, D., Boyen, X., Goh, E.-J.: Hierarchical identity based encryption with constant size ciphertext (2005), http://eprint.iacr.org/2005/015
5. Boneh, D., Gentry, C., Waters, B.: Collusion resistant broadcast encryption with short ciphertexts and private keys. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 258–275. Springer, Heidelberg (2005)
6. Boneh, D., Hamburg, M.: Generalized identity based and broadcast encryption schemes. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 455–470. Springer, Heidelberg (2008)
7. Boneh, D., Sahai, A., Waters, B.: Fully collusion resistant traitor tracing with short ciphertexts and private keys. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 573–592. Springer, Heidelberg (2006)
8. Boyen, X., Waters, B.: Anonymous hierarchical identity-based encryption (without random oracles). In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 290–307. Springer, Heidelberg (2006)

9. Canetti, R., Halevi, S., Katz, J.: A forward-secure public-key encryption scheme. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 255–271. Springer, Heidelberg (2003)
10. Canetti, R., Halevi, S., Katz, J.: A forward-secure public-key encryption scheme. J. Cryptology 20(3), 265–294 (2007)
11. Delerablée, C.: Identity-based broadcast encryption with constant size ciphertexts and private keys. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 200–215. Springer, Heidelberg (2007)
12. Dodis, Y., Fazio, N.: Public key broadcast encryption for stateless receivers. In: Feigenbaum, J. (ed.) DRM 2002. LNCS, vol. 2696, pp. 61–80. Springer, Heidelberg (2003)
13. Gentry, C., Halevi, S.: Hierarchical identity based encryption with polynomially many levels. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 437–456. Springer, Heidelberg (2009)
14. Gentry, C., Silverberg, A.: Hierarchical ID-based cryptography. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 548–566. Springer, Heidelberg (2002)
15. Gentry, C., Waters, B.: Adaptive security in broadcast encryption systems (with short ciphertexts). In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 171–188. Springer, Heidelberg (2009)
16. Goodrich, M.T., Sun, J.Z., Tamassia, R.: Efficient tree-based revocation in groups of low-state devices. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 511–527. Springer, Heidelberg (2004)
17. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: ACM Conference on Computer and Communications Security, pp. 89–98 (2006)
18. Halevy, D., Shamir, A.: The LSD broadcast encryption scheme. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 47–60. Springer, Heidelberg (2002)
19. Jho, N.-S., Hwang, J.Y., Cheon, J.H., Kim, M.-H., Lee, D.H., Yoo, E.S.: One-way chain based broadcast encryption schemes. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 559–574. Springer, Heidelberg (2005)
20. Lewko, A.B., Waters, B.: Fully secure hibe with short ciphertexts, `http://eprint.iacr.org/2009/`
21. Liu, Y.-R., Tzeng, W.-G.: Public key broadcast encryption with low number of keys and constant decryption time. In: Cramer, R. (ed.) PKC 2008. LNCS, vol. 4939, pp. 380–396. Springer, Heidelberg (2008)
22. Naor, D., Naor, M., Lotspiech, J.: Revocation and tracing schemes for stateless receivers. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 41–62. Springer, Heidelberg (2001)
23. Ostrovsky, R., Sahai, A., Waters, B.: Attribute-based encryption with non-monotonic access structures. In: ACM Conference on Computer and Communications Security, pp. 195–203 (2007)
24. Sahai, A., Waters, B.: Revocation systems with very small private keys, `http://eprint.iacr.org/2008/309`
25. Wang, P., Ning, P., Reeves, D.S.: Storage-efficient stateless group key revocation. In: Zhang, K., Zheng, Y. (eds.) ISC 2004. LNCS, vol. 3225, pp. 25–38. Springer, Heidelberg (2004)
26. Waters, B.: Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (2009)
27. Yoo, E.S., Jho, N.-S., Cheon, J.H., Kim, M.-H.: Efficient broadcast encryption using multiple interpolation methods. In: Park, C.-s., Chee, S. (eds.) ICISC 2004. LNCS, vol. 3506, pp. 87–103. Springer, Heidelberg (2005)