Hybrid Broadcast Encryption and Security Analysis

Shaoquan Jiang and Guang Gong

Department of Electrical and Computer Engineering
University of Waterloo
Waterloo, Ontario N2L 3G1, CANADA
Email:{jiangshq, ggong}@calliope.uwaterloo.ca

Jan 30, 2004

Abstract. A broadcast encryption scheme for stateless receivers is a data distribution method which never updates users' secret information while in order to maintain the security the system efficiency decreases with the number of revoked users. Another method, a rekeying scheme is a data distribution approach where it revokes illegal users in an *explicit* and *immediate* way whereas it may cause inconvenience for users. A hybrid approach that appropriately combines these two types of mechanisms seems resulting in a good scheme. In this paper, we suggest such a hybrid framework by proposing a rekeying algorithm for subset cover broadcast encryption framework (for stateless receivers) due to Naor et al. Our rekeying algorithm can simultaneously revoke a number of users. As an important contribution, we formally prove that this hybrid framework has a pre-CCA like security, where in addition to pre-CCA power, the adversary is allowed to *adaptively* corrupt and revoke users. Finally, we realize the hybrid framework by two secure concrete schemes that are based on complete subtree method and Asano method, respectively.

1 Introduction

Broadcast encryption is a mechanism that allows one party to securely distribute his data to privileged users. Since its invention by Fiat and Naor [8], it has been extensively studied [2, 4–6, 10, 14].

A subset cover method based broadcast encryption scheme for stateless receivers was studied by Naor, et al. [12]. Further work appeared in [1, 6, 14]. In this mechanism, a user's secret information is never updated, and user revocation is implicitly achieved by subset cover technique in the broadcast phase. This method has an advantage of no key updating, while it has a drawback that when the number of revoking users grows large, the system efficiency decreases. For example, it takes more time to compute ciphertext, diminishes the effective capacity of users and adds burdens to system management. A complementary mechanism is a rekeying scheme [4, 13, 15, 16] where a user's secret information is explicitly updated for each membership updating. Thus it avoids the weakness of a stateless scheme. However, it may cause inconvenience to users due to frequent membership updatings.

Since the above two mechanisms have complementary features, a hybrid scheme inheriting advantages of two seems to be a good solution. Such a scheme should require the possibility to update each user's secret information. Although this is not the case for applications like DVD, it is absolutely reasonable for applications such as stock quotes, online database, etc. Thus in the sequel, we assume that this condition is always satisfied. The first work along this hybrid approach was due to Garay, et al. [9]. In their method, implicit revocation is achieved by the threshold sharing technique and the property of cover-free family. When the number of the (implicit) revoking users reaches the threshold, it updates affected users' secret information explicitly by uni-cast approach, which is inefficient. Furthermore, a provable security is not provided in this paper. The first provably hybrid scheme was a public key based method due to Dodis, et al [7]. However, in their scheme, a communication overhead is always the security threshold even if there is only one illegal user. Also, to rekeying the users' key set, it only broadcasts a new common secret to them. As a result, leaking such a seed would be possible to enable a revoked user to decrypt the ciphertext again while the traitor is never traced.

In this paper, we propose a hybrid framework called $\mathcal{H}yb$. We obtain this framework by proposing a rekeying algorithm to the subset cover framework for stateless receivers [12]. Our rekeying algorithm can revoke a number of users simultaneously. Furthermore, our algorithm is generic thus is applicable to a wide class of key structure including Logical Key Hierachy (LKH) and it also maintains the key structure of the user's secret information set after updating (e.g. dependence between keys can be used to reduced the user key size, see $\mathcal{H}yb_A$ in Section 3 for an example). In contrast, a simutaneously rekeying algorithm was previously proposed in [11]. However, their algorithm is only applicable to LKH structure and no dependence between keys can be used to reduce the user key size.

In the security definition, the adversary is allowed to have a power of chosen ciphertext attack in the preprocessing model (pre-CCA). As an important contribution, we prove that $\mathcal{H}yb$ framework is secure against such a pre-CCA like attack under the standard model, where besides the pre-CCA power, the adversary is also allowed to corrupt and revoke any user, *adaptively.* Since our rekeying algorithm itself is a generic framework and has LKH rekeying scheme as an example, an interesting implication of our security theorem is that a provable security for such rekeying schemes in fact has be obtained. To our best knowledg, even provable security for simple LKH rekeying has not been previously obtained yet.

Finally, we realize $\mathcal{H}yb$ framework by two pre-CCA secure concrete schemes, $\mathcal{H}yb_{cs}$ and $\mathcal{H}yb_A$ that are based on complete subtree method [12] and Asano method [1], respectively. The latter is most interesting since it demonstrates the rekeying algorithm has a short ciphertext while maintaining the user's key structure. To explictly revoke r users, $\mathcal{H}yb_A$ only needs a length of updating ciphertext proportional to $\frac{r-1}{a-1} - 1 + ar\log_a(n/r)$, where n is the maximal number of users and a is a constant. Since we always set r upbounded by a constant, the overhead above is acutually only logarithmic.

This paper is organized as follows. In section 2 we introduce $\mathcal{H}yb$ method. In section 3 we give two schemes based on complete subtree method and Asano method, respectively. The security of this method is proved in section 4. We end with some discussions in section 5.

2 A Framework for Hybrid Broadcast Encryption

In this section, we suggest a framework for hybrid broadcast encryption that captures the advantages of a stateless scheme and a rekeying scheme both by extending the subset cover framework for stateless receivers by Naor et al. [12]. Our contribution here is mainly a new rekeying algorithm. To achieve this, we explicitly define a user secret information $I(u)$ instead of an abstract symbol in [12]. We call this framework $\mathcal{H}yb$.

Preprocessing Phase

1. Let U be the set of all possible IDs. Broadcast Center (BC) defines a collection of subsets of $U : S_1, \cdots, S_z$, associates a master key I_i and a secret key k_i for $S_i, i = 1, \cdots, z$, where z is polynomially bounded. Suppose that each singleton $\{u\}$ is contained in the collection. (Note: to enable implicitly revoking any subset of users in the broadcast phase. This is necessary. See the

broadcast phase.) I_i implies k_i (and probably also implies some I_j with $S_j \supseteq S_i$). For security reason, we require that I_i is not implied by I_j for any $S_j \supset S_i$. (see the decryption phase.)

2. Let $I(u)$ be the subset of $\{I_i | u \in S_i, i = 1, \cdots, z\}$ obtained by removing all I_t that are implied by another master key, say I_i. We stress that the user key information is defined to consist of some I_i's instead of k_i's since the dependence in the former case will enable to reduce the key size. This is well explained by $\mathcal{H}yb_A$ in Section 3.

Note: throughout this paper, $A \supset B$ means that A strictly contains B. Similar definition is applied to $\subset$.

Join Phase When a new person wants to join, BC first checks whether there is a free ID. If yes, he assigns this ID, say u, together with secret key information $I(u)$ to this person. Later, we refer this person by user u as long as he is not explicitly purged from the system.

Broadcast Phase When BC wants to broadcast message M to all users U except those in R, he first finds a set cover $S_{i_1}, \cdots, S_{i_m}$ such that $S_{i_1} \cup \cdots \cup S_{i_m} = U \backslash R$. Then he forms the ciphertext as

$$\mathcal{H}(M, R) := \langle i_1, \cdots, i_m, E_{k_{i_1}}(k), \cdots, E_{k_{i_m}}(k), F_k(M) \rangle, \tag{1}$$

where E and F are two encryption algorithms and k is a random number of appropriate length. (Note: if the scheme is enabled to *implicitly* revoke any subset of U, then each $\{u\}$ has to be contained in the collection $S_1, \cdots, S_z$ since otherwise there is no way to form a subset cover for the case $U \backslash R = \{u\}$.)

Decryption Phase When $u \in U \backslash R$ receives $\mathcal{H}(M, R)$, he first finds j such that $u \in S_{i_j}$, then he computes k_{i_j} by using $I(u)$ and gets M from it. (Note: If I_i is implied by I_j for some $S_j \supset S_i$, then for $R = U \backslash S_i$, $\mathcal{H}(M, R)$ can be decrypted by an unprivileged user $u \in S_j \backslash S_i$. Thus it is necessary to require that I_i should not be implied by I_j for $S_j \supset S_i$.)

Rekeying Phase In this part, we propose a rekeying algorithm that updates legal users' secret information in order to *explicitly* revoke some users.

Definition 1. *Let $S_1, \cdots, S_z$ be defined as before. We say that S_i has a level l if there exists a chain of length l:*

$$S_{i_1} \subset S_{i_2} \subset \cdots \subset S_{i_{l-1}} \subset S_i,$$

where $i_1, \cdots, i_{l-1}, i$ are distinct; and there exists no such a chain of length $l+1$.

Definition 2. *For two subsets S_i and S_j with $S_i \subset S_j$, if there is no S_t such that $S_i \subset S_t \subset S_j$, then we say that S_i is a child of S_j.*

Let $I_1, \cdots, I_z$ be defined as before. We partition them into subsets $C_1, \cdots, C_\mu$, for some integer μ such that each C_i is generated independently of the rest subsets and no C_i can be further partitioned to smaller such subsets. It follows that if C_i is defined as the output of an algorithm G_i with random input string cn_i, then cn_i is independent of the rest cn_j's. We now define an equivalent relation on $I_1, \cdots, I_z$. We say that I_i, I_j are equivalent if there exists a sequence $I_{l_1} (= I_i), I_{l_2}, \cdots, I_{l_t} (= I_j)$ such that generation procedures for any adjacent keys $I_{l_e}, I_{l_{e+1}}$ partially share random input string. It is clear from the definition of C_i that each C_i is an equivalent class that is independent of the rest C_j's. We let $C(I_i)$ denote the class C_j with $I_i \in C_j$.

Definition 3. *We say that I_i is dominated by $R \subseteq U$ if there exists $I_j \in I(u)$ for some $u \in R$ such that $C(I_i) = C(I_j)$. In this case, we also say S_i is dominated by R. Define*

$$D(R) = \{S_i | I_i \text{ is dominated by } R \text{ and } I_i \in I(u) \text{ for some } u \in U\ \}. \tag{2}$$

We warn that $D(R)$ is not necessary to be simply the collection of all subsets dominated by R.

In order to update I_i to achieve revocation while maintaining the user key dependence so as to keep a small key size, it is necessary to update $C(I_i)$ (i.e., generate fresh I'_j for each $I_j \in C(I_i)$ and make it known to its legal users). Our security theorem in Section 4 implies that this is suficient too. Thus to revoke all the users in R, it is sufficient and necessary to update $\{I_i | S_i \in D(R)\}$. In the following, we present our new simoutaneously rekeying algorithm in Table 1 to achieve this goal where we suppose that the maximal level for $S_1, \cdots, S_z$ is L.

1. BC first determines $D(R)$ and computes new I'_i for all $S_i \in D(R)$;
2. For each $S_i \in D(R)$ at level 1 do
   Suppose $S_i = \{u\}$. If $u \notin R$, then send $E_{k_i}(I'_i)$ to user u.
3. For $l = 2, \cdots, L$ do
   For each $S_i \in D(R)$ at level l do
   For each child S_j of S_i broadcast $E_{k'_j}(I'_i)$ to all users in S_j, where $k'_j = k_j$ if I_j is not updated; otherwise k'_j is the new value.
4. Set IDs in R to be free.

Table 1. Generic Rekeying Algorithm

To get better understanding, a graphic interpretion is demonstrated in Figure 1. There, suppose $R = \{u_1, u_4\}$, $D(R)$ is the collection of all shaded S_i excluding S_{12}. S_{12} has S_9 and S_{10} as its children but I_9, I_{10} imply I_{12}, respectively. Thus, there is no need to transmit I'_{12} to S_9 and S_{10}. Since $\{u_2\}$ and $\{u_5\}$ are both in $D(R)$. Thus, u_2 (resp. u_5) can update I_2 to I'_2 (resp. I_5 to I'_5) while u_1 and u_4 can not update their key information at level 1 (and keys in upperlevels).

Lemma 1.
1. *Every set at level 1 has a form of $\{u\}, u \in U$.*
2. *All users not in R can update his secret information properly.*

Proof. 1. This is an immediate consequence of the fact that any $\{u\}$ is in the subset collection.
2. We only need to show that any new information I'_i for any $I_i \in I(u)$ for some $u \in U$ which is dominated by R can be received by its desired users $S_i \backslash R$. By definition, if $I_i \in I(u)$ is dominated by R, then $S_i \in D(R)$. Thus I_i will be updated to I'_i by Step 1. To show the completeness, we only need to show that for each $S_i \in D(R)$, I'_i can be received by $S_i \backslash R$. This is done by induction on level l. When $l = 1$, S_i has a form of $\{u\}$. By Step 2, if $u \notin R$, then he can get I'_i since he can compute k_i. Assume that for any $S_i \in D(R)$ at level lower than l, its legal users can receive I'_i. We show that for any $S_i \in D(R)$ at level l, its legal users can receive I'_i too. Indeed, for each child S_j of S_i, S_j has a level lower than l. Thus if $S_j \in D(R)$, all users in $S_j \backslash R$ can compute the new version I'_j. If $S_j \notin D(R)$ but dominated by R, by definition of $I(u)$, for each $u \in S_j \backslash R$, there exists an $I_{j'}$ that implies I_j for some $S_{j'}$ with lower level than S_j. Therefore, I'_j can be computed by u. Thus he

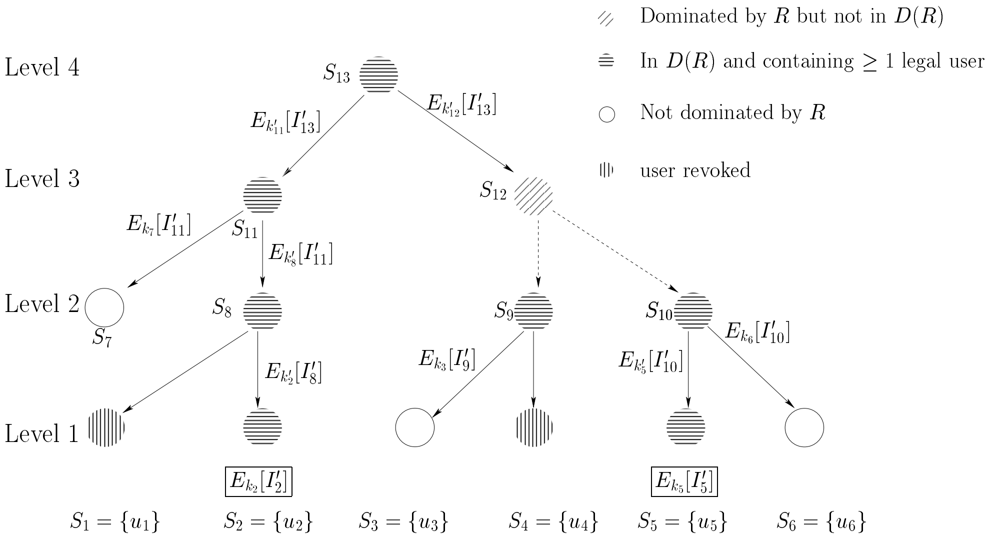

Fig. 1. Graphic Interpretion of Generic Rekeying algorithm

can obtain I'_i. If I_j is not dominated by R at all, then $k'_j = k_j$. Thus $S_j \backslash R$ can obtain I'_i too. On the other hand, for any $u \in S_i$, there exists a child S_j of S_i that contains u since $\{u\}$ is contained in the subset collection. Thus I'_i can be received by $S_i \backslash R$. $\square$

3 Two Concrete Schemes

3.1 $\mathcal{H}yb_{cs}$ scheme

Now we realize the $\mathcal{H}yb$ framework by a concrete construction $\mathcal{H}yb_{cs}$ scheme. This scheme is based on a complete subtree method for stateless receivers [12]. The rekeying algorithm here appeared in [11]. We present it since it is simple. A more interesting scheme is our consequent $\mathcal{H}yb_A$.

Preprocessing Phase

1. BC builds a binary complete tree TR with n leaves. Let these leaves from left to right be users $u_1, \cdots, u_n$. And let the internal nodes be $v_1, \cdots, v_{n-1}$ in width first order. For simplicity, we also identify node u_i with $v_{i+n-1}, i = 1, \cdots, n$. Define S_i to be the set of users rooted at node $v_i, i = 1, \cdots, 2n-1$. BC picks a secret random number k_i of appropriate length and associates it to $S_i, i = 1, \cdots, 2n-1$. Define I_i simply to be k_i.
2. $I(u)$:=$\{I_i | u \in S_i, i = 1, \cdots, 2n-1\}$. In other words, $I(u)$ is the set of k_i lying on the path from u to the root.

Join Phase The same as in the framework.

Broadcast Phase If BC wants to broadcast message M to all users U excluding R, then BC first finds a Steiner tree $Steiner(R)$ (i.e., the smallest subtree of TR that covers users R and the root v_1). Let $v_{i_1}, v_{i_2}, \cdots, v_{i_m}$ be all the nodes that hang off $Steiner(R)$. Then since $S_{i_1} \cup S_{i_2} \cup \cdots \cup S_{i_m} = U \backslash R$, BC forms the ciphertext as follows

$$\mathcal{H}(M,R) = \langle i_1, i_2, \cdots, i_m, E_{k_{i_1}}(k), \cdots, E_{k_{i_m}}(k), F_k(M)\rangle, \tag{3}$$

Decryption Phase When receiving $\mathcal{H}(M,R)$, a user $u \in U \backslash R$ first finds j such that $u \in S_{i_j}$. Since u has k_{i_j} he can get message M.

Rekeying Phase The maximal level among that of subsets $S_1, \cdots, S_{2n-1}$ is $L = 1 + \log n$. For each internal node j with two children j_1, j_2, we have that S_j has exactly two children: S_{j_1}, S_{j_2}. Since that S_i has level l is equivalent to say v_i at depth $L - l$, where the depth of a node is defined as the distance from the root to this node, the rekeying algorithm can be written in Table 2. This algorithm was essentially proposed by Kurnio, et al. [11] which is an extension of that in [3, 15] to achieve simultaneous revocations. Suppose that R is the set of users to be revoked.

1. BC finds $Steiner(R)$ in TR.
2. For each $v_i \in Steiner(R)$ at depth $L-1$,
   BC updates k_i to a random number k_i' of the same length.
3. For $j = L-2, \cdots, 0$
   For each node $v_i \in Steiner(R)$ at depth j,
      BC updates k_i to a random key k_i' of the same length;
      let v_{i_1} and v_{i_2} be the two children of v_i, then
      sends $E_{k'_{i_1}}(k_i')$ to all users rooted at v_{i_1}; sends $E_{k'_{i_2}}(k_i')$ to all users rooted at v_{i_2}, where k'_{i_1} (reps. k'_{i_2}) is the current associated random number for v_{i_1} (resp. v_{i_2}) if it is updated; otherwise, $k'_{i_1} = k_{i_1}$ (reps. $k'_{i_2} = k_{i_2}$).
4. BC sets IDs in R to be free.

Table 2. Rekeying Algorithm for $\mathcal{H}yb_{cs}$

A small example is demonstrated at Figure 2. There, $R = \{u_2, u_5\}$.

Now we briefly discuss the performance of $\mathcal{H}yb_{cs}$. Each user has a key size $|I(u)| = 1 + \log n$. To implicitly revoke r users in the broadcast phase, communication overhead has a upperbound $r \log(n/r)$, which was proved in [12]. To explicitly revoke r users in the rekeying phase, the number of ciphertexts required is upperbounded by $3r - 2 + 2r\log(n/r)$, where the proof is essentially to show that the number of internal nodes in $Steiner(R)$ is upperbounded by $r - 1 + r\log(n/r)$.

3.2 $\mathcal{H}yb_A$ Scheme

In this subsection, we realize $\mathcal{H}yb$ framework by an interesting scheme called $\mathcal{H}yb_A$ scheme. This scheme is based on a subset cover scheme for stateless receivers, which we call Asano method [1]. Our main contribution here is an efficient *simultaneous* rekeying algorithm and a formal proof of the security.

Preprocessing Phase

1. BC chooses a RSA composite $N = pq$ and $2^a - 1$ primes P_h for $h \in \{0,1\}^a \backslash \{0\}$, where p, q are two large primes and a is a constant number. Then he makes N and all P_h for $h \in \{0,1\}^a \backslash \{0\}$ public.

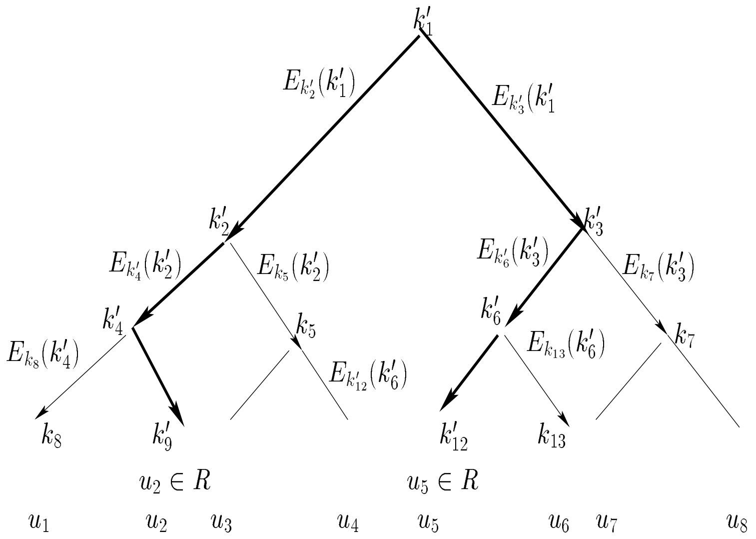

Fig. 2. A Small Example of Rekeying algorithm in $\mathcal{H}yb_{cs}$

2. BC constructs an a-ary complete tree with n leaves. Let these leaves from left to right denote users $u_1, \cdots, u_n$, let the internal nodes be $v_1, \cdots, v_{\frac{n-1}{a-1}}$ in width first order. Identify u_i with $v_{i+\frac{n-1}{a-1}}, i = 1, \cdots, n$. For each $i = 1, \cdots, \frac{n-1}{a-1}$ and $h = h_1 \cdots h_a \in \{0,1\}^a \backslash \{0\}$, let

$$S_{i,h}{:=}\text{set of users rooted at some child } j \text{ (from left to right) of } v_i \text{ for } j \in \{i_1, \cdots i_t\}\ , \tag{4}$$

where $h_l = 1$ for each $l \in \{i_1, \cdots, i_t\}$; $h_l = 0$, otherwise. Let $T_0{=}\prod_{h \in \{0,1\}^a \backslash \{0\}} P_h$. For each internal node v_i, BC chooses a random number k_i and then associates $S_{i,h}$ with key $k_{i,h}{:=}f(k_i^{T_0/P_h})$ and secret information $I_{i,h} = k_i^{T_0/B(h)}$, where $f()$ is a hash function and $B(h){=}\prod_{h \prec b} P_b$. Here "$h \prec b$" means that the ith bit h_i of h is less than or equal to the ith bit b_i of b for all $i = 1, \cdots, a$.
3. Now we define $I(u)$ for a user u as follows

$$I(u){:=}\{I_{i,e_j} | u \text{ is rooted at the } j\text{th child of } v_i, j = 1, \cdots, a, i = 1, \cdots, \frac{n-1}{a-1}\}, \tag{5}$$

where e_j is an a-bit string and each of its component is 0 except the jth bit. For example, if $n = 27$, then $I(u_2) = \{I_{1,100}, I_{2,100}, I_{5,010}\}$ and $I(u_4) = \{I_{1,100}, I_{2,010}, I_{6,100}\}$.

Join Phase User join is done the same as in the framework.

Broadcast Phase If BC wants to broadcast message M to all U except R, then he first finds a Steiner tree $Steiner(R)$ in TR. Let $\{v_{i_1}, v_{i_2}, \cdots, v_{i_m}\}$ be all the internal nodes in $Steiner(R)$. Associate an a-bit number $H(j)$ with each node $v_j \in \{v_{i_1}, \cdots, v_{i_m}\}$, where the tth bit of $H(j)$ is 1 iff the tth child of v_j is not in $Steiner(R)$. Remove v_j from $\{v_{i_1}, \cdots, v_{i_m}\}$ if $H(j) = 0$. WLOG, we still let $v_{i_1}, \cdots, v_{i_m}$ denote the remaining nodes. Then

$$S_{i_1,H(i_1)} \cup \cdots \cup S_{i_m,H(i_m)} = U \backslash R. \tag{6}$$

Thus the ciphertext is defined as follows.

$$\mathcal{H}(M, R){:=}\langle i_1, \cdots, i_m, E_{k_{i_1,H(i_1)}}(k), \cdots, E_{k_{i_m,H(i_m)}}(k), F_k(M)\rangle. \tag{7}$$

For example, consider $n = 27$ and $R = \{u_4, u_{11}\}$ in Figure 3. The $Steiner(R)$ is the thick subtree and the cover subsets are $S_{1,001} = \{u_i | i = 19, \cdots, 27\}$, $S_{2,101} = \{u_1, u_2, u_3, u_7, u_8, u_9\}$, $S_{6,011} = \{u_5, u_6\}$, $S_{3,011} = \{u_{13}, \cdots, u_{20}\}$, $S_{8,101} = \{u_{10}, u_{12}\}$. And the encryption keys used are $k_{1,001} = f(k_1^{T_0/P_{001}})$, $k_{2,101} = f(k_1^{T_0/P_{101}})$, $k_{6,011} = f(k_6^{T_0/P_{011}})$, etc.

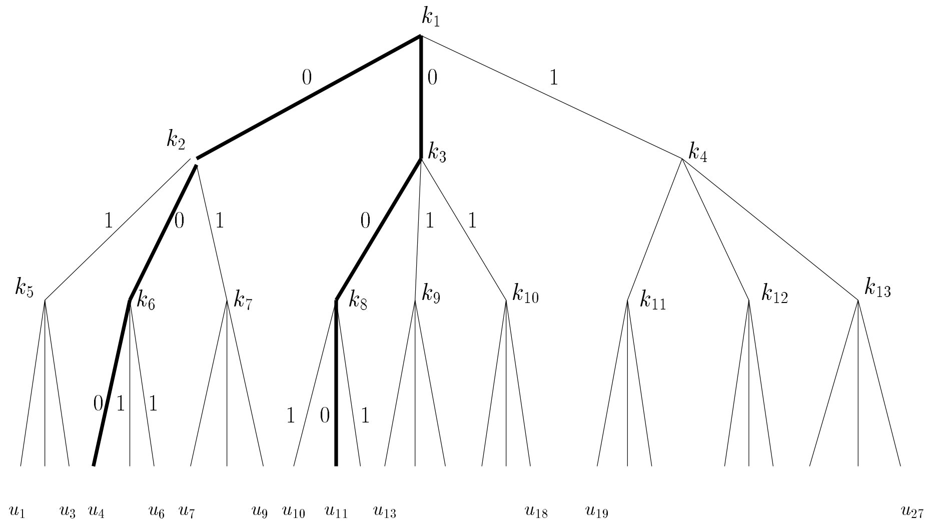

Fig. 3. An Example of Broadcast Procedure in $\mathcal{H}yb_A$

Decryption Phase When receiving $\mathcal{H}(M, R)$, user $u \in U \backslash R$ first finds j such that $u \in S_{i_j, H(i_j)}$. Then he can compute $k_{i_j, H(i_j)}$ from $I_{i_j, e_{j'}}$, where we suppose that u is rooted at the j'th child of v_{i_j}. Then he can get M. In the above example, for $u_1 \in U \backslash R$, we have $I_{2,100} = k_2^{T_0/B(100)} \in I(u_1)$, where $B(100) = P_{100}P_{101}P_{110}P_{111}$. So he can compute $k_{2,101} = f(k_2^{T_0/P_{101}})$ and then decrypt k and M easily.

Rekeying Phase If an internal node v_i is the jth child of another internal node v_t, then $S_{i,e} = S_{t,e_j}$, where a-bit number $e = 11 \cdots 1$. Here when we define notion of *level* and *child* for each $S_1, \cdots, S_z$, we make a little change by "pretending" $S_{i,e}$ is a proper subset of S_{t,e_j}. Under this modification, our rekeying scheme can be adapted from the framework as in Table 3, where R is the set of users to be revoked and thus $D(R) = \{S_{i,e_j} | v_i \in Steiner(R), j = 1, \cdots, a\}$.

For example, consider the example in Figure 3. $R = \{u_4, u_{11}\}$. The main rekeying procedure is as follows.

a. BC updates k_1, k_2, k_3, k_6, k_8 to radom keys $k'_1, k'_2, k'_3, k'_6, k'_8$.
b. **Depth 2:** $E_{k_{6,010}}[I'_{6,010}]$ (resp. $E_{k_{6,001}}[I'_{6,001}], E_{k_{8,100}}[I'_{8,100}], E_{k_{8,001}}[I'_{8,001}]$) is sent to u_5(resp. u_6, u_{10}, u_{12}).
c. **Depth 1:** For node v_2, $E_{k_{5,111}}[I'_{2,100}]$ (resp. $E_{k'_{6,111}}[I'_{2,010}], E_{k_{7,111}}[I'_{2,001}]$) is sent to users rooted at v_5 (resp. v_6, v_7). Similar updating ciphertexts can be computed for node v_3.
d. **Depth 0:** For node v_1, $E_{k'_{2,111}}[I'_{1,100}]$ (resp. $E_{k'_{3,111}}[I'_{1,010}], E_{k_{4,111}}[I'_{1,001}]$) is sent to users rooted at v_2 (resp. v_3, v_4).

1. BC finds Steiner tree $Steiner(R)$,
2. For each node v_i at depth $L-1$ of $Steiner(R)$ (assume the maximal depth is L),
   he changes k_i on node v_i to a random number k_i' of the same length;
   For $j = 1, \cdots, a$, let u be the jth child of v_i, BC sends $E_{k_{i,e_j}}(I'_{i,e_j})$ to u if $u \notin R$, where I'_{i,e_j} is the fresh version of I_{i,e_j}.
3. Define an a-bit number $e = 11 \cdots 1$.
   For $l = L-2, \cdots, 0$ do
   For each node v_i in $Steiner(R)$ at depth l, change k_i to a random number k_i' of the same length.
   For $j = 1, \cdots, a$, do
   Let the jth child of v_i be v_t. Then he broadcasts $E_{k'_{t,e}}(I'_{i,e_j})$ to all users rooted at node v_t, where $k'_{t,e}$ is the new value if it is updated; otherwise $k'_{t,e} = k_{t,e}$. Here I'_{i,e_j} is the fresh version of I_{i,e_j}.
4. BC sets IDs in R to be free.

Table 3. Rekying Algorithm for $\mathcal{H}yb_A$

Now we briefly discuss the performance of $\mathcal{H}yb_A$. The size of a user's personal information $I(u)$ is $\log_a n$. To implicitly revoke r users, the communication overhead in the broadcast phase is $r(1+\log_a(n/r))$, as proved in [1]. To explicitly revoke r users by rekeying algorithm, the number of the required ciphertexts is upperbounded by $\frac{r-1}{a-1} - 1 + ar\log_a(n/r)$, where the proof is essentially to show that the number of internal nodes in $Steiner(R)$ is upperbounded by $\frac{r-1}{a-1} + r\log_a(n/r)$. We stress that if we directly update all secret information dominated by R using KLH, the required ciphertexts would be approximately $2^a/a$ times larger than ours!!

4 Security

In this section, we provide a proof of the security for $\mathcal{H}yb$ method. We first introduce the notion of key indistinguishability which is a variant of that in [12]. Our definition is to use more information about user secret information $I(u)$.

Definition 4. *Let $S_1, S_2, \cdots, S_z$ be defined as before. Consider the key assignment for C_i. Let $\mathcal{B}$ be a probabilistic polynomial time adversary that chooses $I_j \in C_i$ as his target and receives I_t for all $I_t \in C_i$ with $S_t \not\subseteq S_j$. We say that key assignment C_i satisfies key indistinguishability if $\mathcal{B}$ can not distinguish k_j from a random value r_j of the same length, i.e.*

$$|\Pr[\mathcal{B}(A_j, k_j) = 1 \text{ for } j \leftarrow \mathcal{B}] - \Pr[\mathcal{B}(A_j, r_j) = 1 \text{ for } j \leftarrow \mathcal{B}]| \tag{8}$$

is negligible, where $A_j = \{I_t | I_t \in C_i, S_t \not\subseteq S_j\}$.

We say that the (static) key assignment of $\mathcal{H}yb$ framework satisfies key indistinguishability if C_i satisfies this property for each $i = 1, \cdots, \mu$.

Lemma 2. *let $S_1, \cdots, S_z$ be defined as before. Suppose $C_i, i = 1, \cdots, \mu$ satisfies key indistinguishability. Let $S_{i_1}, \cdots, S_{i_m}$ be all the subsets contained in S_j such that $I_{i_t} \in C(I_j), t = 1, \cdots, m$. Then $\langle k_{i_1}, \cdots, k_{i_m}\rangle$ is indistinguishable for any probabilistic polynomial time adversary that receives all I_t for $I_t \in C(I_j)$ with $S_t \not\subseteq S_j$.*

The proof of the lemma is similar to that of Lemma 9 in [12]. So we omit it here.

Now we define the security of a $\mathcal{H}yb$ scheme. This definition captures the threats from explicitly revoked users, current legal users and their collusions. The adversary can schedule any corruption, revocations of users of his choice and he also has a pre-CCA power to request encryption/decryption of broadcast messages/ciphtexts of his choice. Formally,

Definition 5. *Consider the following game between a challenger and an adversary $\mathcal{A}$ against a $\mathcal{H}yb$ scheme.*

1. *$\mathcal{A}$ can take the following actions:*
   (i) He can choose (M_i, R_i) of his choice and request for a ciphertext $\mathcal{H}(M_i, R_i)$;
   (ii) He can ask for decryption of any ciphertext $\mathcal{H}(M_i', R_i')$ of his choice. As a result, he will receive the plaintext M_i';
   (iii) He can request rekeying algorithm on a set R_i' of his choice;
   (iv) He can corrupt any user u. And if a user u is corrupted, then $I(u)$ is provided to $\mathcal{A}$.
2. *Suppose the set of users Ω are currently corrupted (still privileged). Then $\mathcal{A}$ chooses (M, R) of his choice with $\Omega \subseteq R$ and gives it to the challenger.*
3. *The challenger picks $M' = M$ or a random string of the same length and forms a ciphertext $\mathcal{H}(M', R)$. Then he provides it to $\mathcal{A}$, who tries to guess which is the case.*

Then $\mathcal{A}$ outputs a guess bit. $\mathcal{A}$ is said to be successful if his guess is correct. The $\mathcal{H}yb$ scheme is said to be secure *if the success probability of $\mathcal{A}$ is negligible.*

In the above definition, we do not authorize the adversary to control the join operation since this does not result in a higher security. Indeed, our definition does not restrict the join activity of potential users. Thus it contains the case where every user ID is always in use. Especially, if a user is purged from the system, another person will join as this ID immediately. Note security in this case implies the security in other cases no matter the adversary controls the join operation or not since its view of the former covers the view of the latter.

We show that under the above adversary model, our $\mathcal{H}yb$ framework is secure in the standard model. The proof is quite long. We put it in appendix.

Theorem 1. *Assume that the key assignment on C_i satisfies key indistinguishability for $i = 1, \cdots, \mu$, that encryption algorithm E is pre-CCA secure, and that F is semantically secure. Then the $\mathcal{H}yb$ framework is* secure.

Now let us discuss the security of concrete schemes $\mathcal{H}yb_{cs}$ and $\mathcal{H}yb_A$.

Lemma 3. *For $\mathcal{H}yb_{cs}$ scheme, we have $C(I_i) = \{I_i\}$ and key assignment indistinguishability holds for $C(I_i), i = 1, \cdots, 2n-1$.*

Proof Since each k_i is uniformly random, it follows $C(I_i) = \{I_i\}$. Key indistinguishability holds since $C(I_i)$ is not dominated by $U \backslash S_i$. □

By using Theorem 1, we have

Corollary 1. *If encryption algorithm E is pre-CCA secure and F is semantically secure, then $\mathcal{H}yb_{cs}$ is secure.*

For $\mathcal{H}yb_A$ scheme we first have the following lemma.

Lemma 4. *For each i and a non-zero a-bit string h, $C(I_{i,h}) = \{I_{i,b} | b \in \{0,1\}^a \backslash \{0\}\}$. And if we assume $f()$ is a random oracle, then key assignment indistinguishability holds for $C(I_{i,h})$.*

Proof The first conclusion follows from the fact: k_i is uniform and independent of $\{k_j | j \neq i\}$. Now we show the key assignment indistinguishability of $C(I_{i,h})$ holds. For given i, $S_{i,b} \subseteq S_{i,h}$ if and only if $b \prec h$. Thus for an adversary $\mathcal{B}$ that attempts to break the key indistinguishability of $C(I_{i,h})$, he will receive $I_{i,b}$ for all b such that $b \not\prec h$, *i.e.*, $\exists t$ s.t. $h_t = 0$ and $b_t = 1$, where h_t (resp. b_t) is the tth bit of h (resp. b). Notice the following fact: assume that k is a random number and a, c are two numbers less than N. Assume that $\gcd(a, c) = d$. Then for given $k^a \pmod N$ and $k^c \pmod N$, one can compute $k^d \pmod N$ in $O(\log^3 N)$.
This fact can be easily proved by using the Euclidean algorithm. Now we come back to our proof. Notice $I_{i,b} = k_i^{T_0/B(b)}$. It follows from the above fact that for given $I_{i,b}$ for all b with $b \not\prec h$, one can efficiently compute $k_i^{\frac{T_0}{LCM\{B(b)|b \not\prec h\}}}$, which in fact is $k_i^{\prod_{0 \neq b \prec h} P_b}$. Here $LCM()$ is the least common multiple function. On the other hand, for given $k_i^{\prod_{0 \neq b \prec h} P_b}$, one can easily compute $I_{i,b}$ for all b with $b \not\prec h$. Thus we only need to show that for given $k_i^{\prod_{0 \neq b \prec h} P_b}$, $k_{i,h}$ is indistinguishable to $\mathcal{B}$. Actually, we show that if there exists algorithm $\mathcal{B}$ that distinguishes $k_{i,h}$ from a random string of the same length with non-negligible advantage, then there exists an algorithm $\mathcal{I}nv$ that inverts RSA function x^{P_h} with non-negligible probability. Now upon input $\alpha = x^{P_h}$, $\mathcal{I}nv$ does the following

0. $\mathcal{I}nv$ finds Q_1 and Q_2 efficiently such that $Q_1 P_h + Q_2 T_0/P_h = 1$ by using the Euclidean algorithm.
1. $\mathcal{I}nv$ chooses a random number r and computes $\beta = \alpha^{(\prod_{0 \neq b \prec h} P_b)/P_h}$. Then he provides β together with r to $\mathcal{B}$.
2. To answer $\mathcal{B}$'s queries for $f()$ function, $\mathcal{I}nv$ maintains a $f-$list. Initially, this list is empty. For each query y_i, $\mathcal{I}nv$ checks in $f-$list whether y_i was queried before. If yes, he provides the answer recorded in the list to $\mathcal{B}$. Otherwise, $\mathcal{I}nv$ computes $\gamma_i := \alpha^{Q_1} \cdot y_i^{Q_2}$ and checks whether $\gamma_i^{P_h} = \alpha$. If yes, he announces success and outputs γ_i. If y_i is not queried before, he chooses a random number g_i of length l and provides it to $\mathcal{B}$, where l is the output length of f. At the same time, he adds the pair (y_i, g_i) into his $f-$list.
3. Finally, $\mathcal{B}$ outputs a bit b' for a guess whether his challenge is random or not. If $\mathcal{I}nv$ does not announce for success in the experiment, then he quits with failure.

First, $\beta = x^{\prod_{0 \neq b \prec h} P_b}$. Since $f()$ is a random oracle, it follows that (β, r) is distributed the same as in the real world. By calculation, we can verify $\gamma_i^{P_h} = \alpha$ if and only if $y_i = x^{T_0/P_h}$. Thus the responses to queries from $\mathcal{B}$ are distributed the same as the responses from $f()$ oracle of $\mathcal{B}$. If x^{T_0/P_h} is not queried before he outputs the guess bit, the guess b' is correct with probability exactly 1/2. Assume the probability that x^{T_0/P_h} is queried is ϵ, then the advantage of $\mathcal{B}$ is at most $\epsilon + \frac{1-\epsilon}{2} - \frac{1-\epsilon}{2} = \epsilon$. Since we assume his advantage is non-negligible, it follows that ϵ is non-negligible. On the other hand, once x^{T_0/P_h} is queried, $\mathcal{I}nv$ will succeed. It follows that the success probability of $\mathcal{I}nv$ is ϵ, non-negligible, a contradiction to the hardness of inverting RSA function. □

Now we investigate the security $\mathcal{H}yb_A$. Since we have made a modification on the definition for *level* and *child*, we can not directly apply Theorem 1. However, one can check that the proof of the security theorem still goes through without any modification. Thus we have

Corollary 2. *If encryption algorithm E is pre-CCA secure, F is semantically secure and $f()$ is a random oracle, then $\mathcal{H}yb_A$ scheme is secure.*

5 Discussions

In this section, we give some discussions.

1. **On independence of $C_1, \cdots, C_\mu$.** Previously, we suppose the random bits used to generate each $C(I_j)$ are independent of anything else. In reality, to flip a long sequence of random bits in order to satisfy this condition is not practical. However, we stress that in fact this is not necessary. We can replace the long sequence of coin flips by a pseudorandom sequence. And the security of this framework still holds if the original version is secure. The proof is by standard argument. Specifically, if the security is compromised due to this replacement, then we can distinguish this pseudorandom sequence from a random sequence of the same length.
2. **Traceability.** Traitor tracing is to find out the illegal users that help construct a pirate decoder. In [12], Naor, et al. proposed a binary search like tracing algorithm. Since $\mathcal{H}yb$ method is also based on subset-cover method, it follows that their tracing algorithm is applicable if the considered scheme is secure and a bifurcation property is satisfied.
3. **Implication of Secure Rekeying Scheme.** If there is some $S_i = U$, and broadcast encrytion is as $\langle E_{k_i}(k), F_k(M)\rangle$, then our rekeying algorithm is in fact a framework for a rekeying scheme. Thus the security theorem implies the security of this framework. Thus, the provable security for the popular LKH rekeying scheme is obtained. To our best knowledge, it is the first formal proof for such a scheme.
4. **On unlimited number of users.** For a fixed subset cover method, the maximal number of users it can support is set in advance. We claim it is easy to obtain a system that supports unlimited number of users. For simplicity, suppose that $\mathcal{T}_i$ is a realization of $\mathcal{H}yb$, which can support 2^i users. We construct a system $\mathcal{T}$ as follows. Initially, $\mathcal{T}$ is set to $\mathcal{T}_0$. When a user joins in, BC first checks whether every user ID in $\mathcal{T}_0$ is in use. If not, it assigns a free ID to the new user. If yes, BC independently generates $\mathcal{T}_1$ and assigns an ID to the new user. At some moment, let $\mathcal{T}$ be composed of $\mathcal{T}_0, \cdots, \mathcal{T}_i$. If at this time, a new user joins in, then BC similarly first tries to find a free ID from $\mathcal{T}_t, t = 0, \cdots, i$. If yes, he assigns a free ID and corresponding secret information to the new user. Otherwise, he independently generates $\mathcal{T}_{i+1}$ and assigns a free ID and secret information to the new user. Broadcast and rekeying operations are done for each $\mathcal{T}_i$ in $\mathcal{T}$ *individually.* For the security, we claim that if $\mathcal{T}_i$ is secure, then $\mathcal{T}$ is pre-CCA too. The proof is by a simple hybrid argument. For the efficiency, if we take $\mathcal{T}_i$ by $\mathcal{H}yb_{cs}$ with maximal number of users 2^i, then communication overhead and cost of rekeying algorithm only additively increase by at most $O(\log n)$. A similar construction is applied to the case $\mathcal{T}_i$ taken as $\mathcal{H}yb_A$ with a maximum a^i users.

References

1. T. Asano, A Revocation Scheme with Minimal Storage at Receivers, *Advanced in Cryptology-Asiancrypt'02,* Y. Zheng (Ed.), LNCS 2501, Springer-verlag, 2002, pp. 433-450.
2. D. Boneh and M. K. Franklin, An Efficient Public Key Traitor Tracing Scheme, *Advances in Cryptology-CRYPTO'99,* M. J. Wiener (ed.), LNCS 1666, Springer-verlag, 1999, pp. 338-353.
3. R. Canetti, J. A. Garay, G. Itkis, D. Micciancio, M. Naor and B. Pinkas, Multicast Security: A Taxonomy and Some Efficient Constructions, *IEEE INFOCOM'99*, 21-25, March 1999, New York, Vol. 2, 708-716
4. R. Canetti, T. Malkin and K. Nissim, Efficient Communication-Storage Tradeoffs for Multicast Encryption, *Advances in Cryptology-EUROCRYPT'99,* J. Stern (Ed.), LNCS 1592, Springer-verlag, 1999, pp. 459-474.
5. B. Chor, A. Fiat and M. Naor, Tracing Traitors, *Advances in Cryptology-CRYPTO'94*, Y. Desmedt (Ed.), LNCS 839, Springer-verlag, 1994, pp. 257-270.
6. Y. Dodis and N. Fazio, "Public Key Broadcast Encryption for Stateless Receivers", *ACM Workshop on Digital Rights Management,* November 2002. Available at http://theory.lcs.mit.edu/~yevgen/academic.html
7. Yevgeniy Dodis, Nelly Fazio, Aggelos Kiayias and Moti Yung, "Scalable Public-Key Tracing and Revoking", Principles of Distributed Computing (PODC), July 2003.
8. A. Fiat and M. Naor, Broadcast Encryption, *Advances in Cryptology-CRYPTO'93,* D. Stinson (Ed.), LNCS 773, Springer-verlag, 1994, pp. 480-491.

9. Juan A. Garay, Jessica Staddon and Avishai Wool, Long-Lived Broadcast Encryption, *Advances in Cryptology-Crypto'00,* M. Bellare (Ed.), LNCS 1880, Springer-verlag, 2000, pp. 333-352.
10. A. Kiayias and M. Yung, Traitor Tracing with Constant Transmission Rate, *Advances in Cryptology-EUROCRYPT'02,* L. R. Knudsen(Ed.), LNCS 2332, Springer-verlag, 2002, pp. 450-465.
11. Hartono Kurnio, Reihaneh Safavi-Naini, Huaxiong Wang: A Secure Re-keying Scheme with Key Recovery Property. ACISP 2002: 40-55.
12. D. Naor, M. Naor and J. Lotspiech, Revocation and Tracing Schemes for Stateless Receivers, *Advances in Cryptology-Crypto'01,* J. Kilian (Ed.), LNCS 2139, Springer-verlag, 2001, pp. 41-62.
13. Jessica Staddon, Sara K. Miner, Matthew K. Franklin, Dirk Balfanz, Michael Malkin and Drew Dean, Self-Healing Key Distribution with Revocation, *IEEE Symposium on Security and Privacy 2002,* May 12-15, 2002, Berkeley, California, USA, pp. 241-257.
14. D. R. Stinson and R. Wei, Combinatorial Properties and Constructions of Traceability Schemes and Frameproof Codes, *SIAM Journal on Discrete Mathematics,* 11(1): 41-53 (1998).
15. D. M. Wallner, E. J. Harder and R. C. Agee, Key Management for Multicast: Issues and Architectures, Internet Request for Comments 2627, June, 1999. Available: ftp.ietf.org/rfc/rfc2627.txt
16. C. K. Wong and M. S. Lam, Secure Group Communication Using Key Graphs, *Sigcomm'98.*

Appendix

Proof of Theorem 1 If The $\mathcal{H}yb$ framework is insecure, then there exists an adversary $\mathcal{A}$ that breaks the security in Definition 5. We show that there is an adversary $\mathcal{B}$ that can break the security of E. $\mathcal{B}$ achieves this by running a simulated copy of the security game in Definition 5 and answering the queries of $\mathcal{A}$ with the help of his challenger. The whole is composed of several lemmas. We first introduce the game on the strategy of $\mathcal{B}$.

1. $\mathcal{B}$ uniformly chooses $j \in \{1, 2, \cdots, z\}$ and t uniformly from $\{1, \cdots, Q\}$, where Q is an upperbound of the number of the cover subsets when computing the ciphertext in the broadcast phase. Let the number of requests of rekeying algorithm on any set R' with $R' \cap S(I_j) \neq \emptyset$ be upperbounded by $\lambda - 1$, where $S(I_j) = \cup_{I_t \in C(I_j)} S_t$. Finally $\mathcal{B}$ chooses d uniformly from $\{0, 1, \cdots, \lambda - 1\}$.
2. $\mathcal{B}$ simulates $\mathcal{H}yb$ scheme with $S_1, \cdots, S_z$ defined before. And then he runs $\mathcal{A}$ against it. We use d' to denote the number of requests up to date for running rekeying algorithm on any set R' with $R' \cap S(I_j) \neq \emptyset$. Initially, $d' = 0$.
3. If $\mathcal{A}$ asks for revoking R'_i with $R'_i \cap S(I_j) \neq \emptyset$, then $\mathcal{B}$ increases d' by $d' = d' + 1$. If $d' > d$, $\mathcal{B}$ aborts. Otherwise, $\mathcal{B}$ uses his own random inputs to generate a fresh copy $C(I'_t)$ for $S_t \in D(R'_i)$ (note if I_t and $I_{t'}$ are within the same class, only one fresh copy $C(I'_t)$ is generated). Then he forms the updating ciphertext of I'_t by using his own knowledge except for the special case $d' = d$. In this case, he first chooses a random number r_w of length $|k_w|$ for each $S_w \subset S_j$ with $I_w \in C(I_j)$. Then if k'_w, satisfying $S_w \subset S_j$ and $I_w \in C(I_j)$, is required as the encryption key, then instead of using k'_w he uses r_w (fixed throughout the case $d' = d$); and if k'_j is required in order to generate a ciphertext of I'_t, then he requests for the ciphertext of I'_t from his encryption oracle. Furthermore, in case for the first time it reaches $d' = d$, if it needs to encrypt I'_w for $S_w \subseteq S_j$, $\mathcal{B}$ encrypts a random string rd_w of the same length instead.

   If $\mathcal{A}$ asks for revoking R'_i with $R'_i \cap S(I_j) = \emptyset$, then d' is kept unchanged. The rest actions are the same as in the case $R'_i \cap S(I_j) \neq \emptyset$ except for the case $d' = d$. In this case, if k'_w, satisfying $S_w \subset S_j$ and $I_w \in C(I_j)$, is required as an encryption key, he uses r_w chosen before; if k'_j is required as an encryption key, then he queries his encryption oracle.
4. If $\mathcal{A}$ asks to corrupt $u \notin S_j$, then $\mathcal{B}$ provides $I(u)$ to $\mathcal{A}$ by using his own knowledge.
   If $\mathcal{A}$ asks to corrupt $u \in S_j$ and $d' < d$, then $\mathcal{B}$ provides $I(u)$ to $\mathcal{A}$ by using his knowledge too.

If $\mathcal{A}$ asks to corrupt $u \in S_j$ and $d' = d$, then $\mathcal{B}$ aborts. (note in case $d' > d$, $\mathcal{B}$ already aborts in Step 3 .)

5. When $\mathcal{A}$ requests encryption/decryption of an arbitrary (M_i, R_i)/ciphertext, $\mathcal{B}$ computes it by using his knowledge if no k_w, satisfying $S_w \subseteq S_j$ and $I_w \in C(I_j)$, is required or if $d' < d$. If $d' = d$ and k_j is required for encryption/decryption, then in case of encryption, he chooses the session k uniformly random of appropriate length and asks for its encryption oracle and in case of decryption, he asks for his decryption oracle. If $d' = d$ and k_w, satisfying $S_w \subset S_j$ and $I_w \in C(I_j)$, is required, then he uses r_w chosen before.
6. Suppose Ω is the set of users currently corrupted (i.e., corrupted but still priviledged) by $\mathcal{A}$. If $\mathcal{A}$ chooses $(M, R), R \supseteq \Omega$ for test, $\mathcal{B}$ finds a subset cover $S_{i_1} \cup S_{i_2} \cup \cdots \cup S_{i_m} = U \backslash R$. If $i_t = j$ and $d' = d$, then $\mathcal{B}$ announces for a test. Otherwise, $\mathcal{B}$ aborts. If $\mathcal{B}$ does not abort, he chooses a random number k of appropriate length and gives it to the challenger. The challenger provides $\alpha \in \{E(k), E(x_t)\}$ randomly to $\mathcal{B}$, where x_t is a random string of length $|k|$. Upon receiving α, $\mathcal{B}$ chooses $M' = M$ or a random string M'' of length $|M|$ equally likely and forms the ciphertext
$$\langle i_1, \cdots, i_m, E_{k_{i_1}}(x_1), \cdots, E_{k_{i_{t-1}}}(x_{t-1}), \alpha, E_{k_{i_{t+1}}}(k), \cdots, E_{k_{i_m}}(k), F_k(M') \rangle, \tag{9}$$
where $x_i, i = 1, \cdots, t-1$ are uniformly random of the length $|k|$. And then $\mathcal{B}$ provides the above ciphertext to $\mathcal{A}$.

If $\mathcal{B}$ does not abort, then in case $M' = M$, $\mathcal{B}$ outputs whatever $\mathcal{A}$ outputs; in case M' is random, $\mathcal{B}$ complements the output of $\mathcal{A}$. If $\mathcal{B}$ aborts somewhere, then it outputs 0, 1 equally likely.

We denote the above game by Γ^{rand}. Note in this game, $\mathcal{B}$, playing the role of the challenger for $\mathcal{A}$, does not strictly follows actions defined in the security defintion. For example, in the special case, r_w is used instead of k'_w; in step 3, the ciphertext for I'_w is changed to the ciphertext for a random string rd_w. We will prove that this change essentially does not matter. Toward this purpose, We define a variant Γ^{real} of game Γ^{rand}. Γ^{real} is the same as Γ^{rand} with exception in the case of revoking some R_i with $R_i \cap S(I_j) \neq \emptyset$ and $d' = d$ in Step 3. In Γ^{real}, instead of generating new $C(I'_j)$ by himself, $\mathcal{B}$ will receive all $I'_t \in C(I'_j)$ for $S_t \not\subseteq S_j$ and furthermore receive k'_w for all $S_w \subset S_j$ with $I'_w \in C(I'_j)$. And he does not need to generate r_w for $S_w \in S_j$ with $I_w \in C(I_j)$ and later when required to use r_w, he uses k'_w that is received above. His encryption/decryption oracle will use the secret key k'_j instead of a random number in Γ^{rand}.

Our plan for the proof of security of $\mathcal{H}yb$ is as follows.

1. The probability that $\mathcal{B}$ won't abort in game Γ^{real} is negligibly close to $\frac{1}{z\lambda}$. And it is negligibly close to the probability in game Γ^{rand}.
2. If an adversary in the $\mathcal{H}yb$ scheme has a non-negligible advantage, then adversary $\mathcal{B}$ in game Γ^{real} has a non-negligible advantage, too.
3. If adversary $\mathcal{B}$ in game Γ^{rand} has a negligible advantage while it has a non-negligible advantage in game Γ^{real}, then there exists an adversary $\mathcal{D}$ that compromises the key assignment indistinguishability of $\mathcal{H}yb$.

Based on the key assignment indistinguishability of $\mathcal{H}yb$ scheme and items 2, 3, we conclude that the pre-CCA security of E is compromised, a contradiction.

Lemma 5. *d' is the number of times that $C(I_j)$ has been updated up to date.*

Proof The proofs for both games Γ^{real} and Γ^{rand} are identical. Note that if $R'_i \cap S(I_j) = \emptyset$, $C(I_j)$ is not dominated by R'_i. Thus $C(I_j)$ keeps unupdated. In this case, d' remains unchanged by description of the game. On the other hand, if $R'_i \cap S(I_j) \neq \emptyset$, then there exists $u \in R'_i \cap S(I_j)$. Thus $C(I_j)$ is dominated by u. Thus $C(I_j)$ will be updated. By the description of the game, $d' = d' + 1$ in this case. □

Define $\mathbf{Non-abort}(\Gamma^c)$ to be the event in game Γ^c in which the adversary $\mathcal{B}$ does not abort, where $c \in \{real, rand\}$. We have the following lemma.

Lemma 6. $\Pr[\mathbf{Non-abort}(\Gamma^{real})] \approx \frac{1}{z\lambda}$, *where* $\approx$ *means "negligibly close".*

Proof Suppose $(S_{i_1}, S_{j_1}), \cdots, (S_{i_q}, S_{j_q})$ are all the possible pairs of subsets, satisfying $S_{i_t} \subseteq S_j$ and S_{i_t} is a child of S_{j_t}, or satisfying $i_t = j_t$ and $S_{i_t} = \{u\}$ for some $u \in S_j$, where $t = 1, \cdots, q$. Suppose these pairs are arranged such that $|S_{j_1}| \leq |S_{j_2}| \leq \cdots \leq |S_{j_q}|$. In game Γ^{real}, when reaching the case $d' = d$ for the first time, $\mathcal{B}$ is supposed to send the rekeying ciphertexts, $E_{k'_{i_t}}(I'_{j_t})$ and actually he sends $E_{k'_{i_t}}(rd_{j_t})$ in game Γ^{real} for $t = 1, \cdots, q$. Note here k'_{i_t} is the currently used key (before receiving this ciphertext), i.e., $k'_{i_t} = k_{i_t}$ if it has not been updated (for example, at level one, we always have $k'_{i_t} = k_{i_t}$.

We then define a sequence of hybrid games of Γ^{real}, which we denote by Γ_l^{real}, where $l = 0, 1, \cdots, q$. Then the main difference between Γ^{real} and Γ_l^{real} is in the above special event when $\mathcal{B}$ is supposed to send $E_{k'_{i_t}}(I'_{j_t}), t = 1, \cdots, q$. In Γ^{real}, $\mathcal{B}$ sends $E_{k'_{i_t}}(rd_{j_t})$ for all $t = 1, \cdots, q$. However, in Γ_l^{real}, $\mathcal{B}$ sends $E_{k'_{i_t}}(rd_{j_t})$ for $1 \leq t \leq l$ and he sends $E_{k'_{i_t}}(I'_{j_t})$ for $l < t \leq q$. Furthermore, to enable him to do this, $\mathcal{B}$ will receive I'_w for all $S_w \not\subseteq S_{i_l}$, together with all k'_r, satisfying $S_r \subseteq S_{j_l}$ and $I_r \in C(I_j)$. Note that $\Gamma_q^{real} = \Gamma^{real}$. And Γ_0^{real} is the game where $\mathcal{B}$ actually know $C(I'_j)$. Since Γ_0^{real} actually does not relate to q, we simply write it as $\Gamma^{real'}$ In the following, we show that the probabilities of non-abort events in $\Gamma^{real'}$ and Γ^{real} are negligibly close. If this were not true, we show that there exists an adversary $\mathcal{D}$ that can compromise the key assignment indistinguishability of $C(I_j)$ for some j. The action of $\mathcal{D}$ can be described as follows.

1. $\mathcal{D}$ chooses j from $\{1, \cdots, z\}$ uniformly and then he selects l uniformly from $\{1, \cdots, \nu\}$, where ν is the upperbound of q. If $l > q$, then $\mathcal{D}$ exits with 1 or 0 equally likely; otherwise, $\mathcal{D}$ announces k'_{i_l} as his target. As a response, he will receive all $I'_w \in C(I'_j)$ for all $S_w \not\subseteq S_{i_l}$ as well as $\alpha_w \in \{k'_w, r_w\}$ for $S_w \subseteq S_{i_l}$, where all are taken from the first component or all are taken from the second componenet and the probability is 1/2.
2. $\mathcal{D}$ follows the decription of Γ_{l-1}^{real} except when reaching the special case $d' = d$ for the first time. In this case, if required to send $E_{k'_{i_l}}(I'_{j_l})$, $\mathcal{D}$ sends $E_{\alpha_{i_l}}(I'_{j_l})$ or $E_{\alpha_{i_l}}(rd_{j_l})$ with probability 1/2 if $|S_{i_l}| > 1$; he sends $E_{\alpha_{i_l}}(rd_{j_l})$ if $|S_{i_l}| = 1$.
3. If $\mathcal{D}$ does not abort, the in case he sent $E_{\alpha_{i_l}}(I'_{j_l})$ in the exception he outputs 1; in case he sent $E_{\alpha_{i_l}}(rd_{j_l})$ in the exception, he outputs 0. If $\mathcal{D}$ does abort, then in case he sent $E_{\alpha_{i_l}}(I'_{j_l})$ in the exception, he outputs 0; in case he sent $E_{\alpha_{i_l}}(rd_{j_l})$ in the exception, he ouputs 1.

Now we calculate the advantage $\mathbf{Adv}(\mathcal{D})$ of $\mathcal{D}$. Let $p_l^1(j)$ be the non-abort probability in game Γ_l^{real}, for a fixed j. Also define $p_l^0(j)$ be the non-abort probability for a fixed j in the avariant game of Γ_l^{real} where in the special case k'_{i_t} is replaced by r_{i_t} for all t satisying $S_{i_t} \subseteq S_j$ and $I_{i_t} \in C(I_j)$.

Define l_j to be the number t such that $|S_{i_t}| = 1$ but $|S_{i_{t+1}}| > 1$ for a fixed choice j. Then we have

$$\begin{aligned}
\mathbf{Adv}(\mathcal{D}) &= \frac{1}{z\nu}\sum_{j=1}^{z}\sum_{l=l_j+1}^{q}\frac{p^0_{l-1}(j)+1-p^0_l(j)}{2} - \frac{1}{z\nu}\sum_{j=1}^{z}\sum_{l=l_j+1}^{q}\frac{p^1_{l-1}(j)+1-p^1_l(j)}{2} \\
&= \frac{1}{2z\nu}\sum_{j=1}^{z}[(p^0_{l_j}(j) - p^0_q(j)) - (p^1_{l_j}(j) - p^1_q(j))] \\
&\approx \frac{1}{2z\nu}\sum_{j=1}^{z}(p^1_{l_j}(j) - p^1_q(j)) \\
&\approx \frac{1}{2\nu}(\Pr[\mathbf{Non-abort}(\Gamma^{real'})] - \Pr[\mathbf{Non-abort}(\Gamma^{real})].
\end{aligned}$$

Here the first $\approx$ holds since $p^0_{l_j}(j) \approx p^0_q(j)$, which can be proved using standard argument to reduce the pre-CCA security of E; the second $\approx$ holds since $p^1_0(j) \approx p^1_{i_1} \approx \cdots \approx p^1_{i_{l_j}}$, which can be proved by noticing the following facts :

- $|S_{i_t}| = 1$ for $t = 1, \cdots, l_j$.
- If there exists two adjacent probabilities $p^1_t(j)$ and p^1_{t+1} with a non-negligible gap, then one can easily compromise the indistinguishability of $k_{i_{t+1}}$ with a non-negligible gap, which can be done by noticing that the rekeying ciphertexts here can be simulated in both of these two games.

Thus we have $\mathcal{D}$ has a non-negligible advantage, contradiction to the assumption of the key assignment indistinguishability of $C(I_j)$ for all $j = 1, \cdots, z$.

Consider a variant $\Gamma^{real''}$ of game $\Gamma^{real'}$. For case $d' = d$ at Step 4, suppose that in game $\Gamma^{real''}$ instead of abortion, $\mathcal{B}$ responses faithfully. He can do this because he knows $C(I'_j)$. The rest of the action is unchanged (although $\mathcal{B}$ can compute k_j already, we consider the case $\mathcal{B}$ still follows its described action. Our point is $\mathcal{A}$ can realize whether $\mathcal{B}$ is normal or not). We show that $\mathcal{B}$ aborts in $\Gamma^{real'}$ if and only if it aborts in game $\Gamma^{real''}$. Suppose x is a transcript in $\Gamma^{real'}$ in which $\mathcal{B}$ aborts at Step 4 and x' is the transcript in $\Gamma^{real''}$ with prefix being x while instead of abortion at Step 4 $\mathcal{B}$ continues his action described above. If $\mathcal{B}$ won't abort in x', then when $\mathcal{A}$ announces for a test by providing (M, R), $d' = d$ since if $d' > d$ then $\mathcal{B}$ will abort at Step 3. It follows that u is not revoked (i.e., currently he is a privileged user). Since we assume that $\mathcal{A}$ is a valid attacker, it follows that $u \notin R$. Thus for any subset cover $S_{i_1} \cup S_{i_2} \cup \cdots \cup S_{i_m} = U \backslash R$, there exists no t such that $i_t = j$. Therefore, $\mathcal{B}$ must abort, a contradiction. Thus $\Pr[\mathbf{Non-abort}(\Gamma^{real'})] = \Pr[\mathbf{Non-abort}(\Gamma^{real''})]$.

Now we consider $\Pr[\mathbf{Non-abort}(\Gamma^{real''})]$. Let $\Pi^{worl}(D)$ denote the set of the views of adversary $\mathcal{A}$ in the real world (i.e. in Defintion 5) with restriction that the number of requests of rekeying algorithm on revoking set R' with $R' \cap S(I_j) \neq \emptyset$ is D. Note that the view of adversary $\mathcal{A}$ in Step 1-5 in game $\Gamma^{real''}$ before his abortion is distributed exactly the same as in the real world since $\mathcal{B}$'s action is according to the real world. If instead of abortion when $d' > d$ at Step 3, $\mathcal{B}$ continues the normal action as described in the real world, the adversary view in Step 1-5 will be distributed exactly the same as in the real world. It follows that given d chosen by $\mathcal{B}$, if $\mathcal{B}$ won't abort in Step 1-5, the view of $\mathcal{A}$ during Step 1-5 is distributed exactly the same as in the real world conditional on $D \le d$, where D is the number defined before. And therefore, the non-abort probability in Step 1-5 in game $\Gamma^{real''}$ is $\sum_{D \le d}\Pr[\Pi^{worl}(D)]$, where $\Pr[\ ]$ is according to distribution of the view of adversary $\mathcal{A}$ in the real world.

Furthermore, in Step 6, since $\mathcal{A}$ is assumed to be valid, it follows that if $\mathcal{B}$ won't abort till $\mathcal{B}$ receives $\mathcal{A}$'s test query (M, R), the adversary view of $\mathcal{A}$ is distributed the same as in the real world conditional on $D \le d$. And since at this point $\mathcal{B}$ won't abort if and only if $i_t = j$ and $d' = d$, it follows that conditional on that $\mathcal{B}$ won't abort, the adversary view till just before he reads the test ciphertext is distributed the same as $x \in \Pi^{worl}_{i_t=j}(d)$ in the real world, where $\Pi^{worl}_{i_t=j}(d)$ is the subset of $\Pi^{worl}(d)$ with the restriction $i_t = j$. Thus given t, we have

$$\begin{aligned}\Pr[\mathbf{Non-abort}(\Gamma^{real''})] &= \tfrac{1}{\lambda}\textstyle\sum_{d=0}^{\lambda-1}\sum_{x\in\Pi^{worl}(d)}\Pr[i_t = j, x]\\ &= \tfrac{1}{\lambda}\textstyle\sum_{\Pi^{worl}}\Pr[i_t = j, x]\\ &= \Pr[i_t = j]\\ &= \tfrac{1}{z\lambda},\end{aligned}$$

where $\Pi^{worl} = \cup_{d=0}^{\lambda-1}\Pi^{worl}(d)$. Therefore, we have $\Pr[\mathbf{Non-abort}(\Gamma^{real'})] = \frac{1}{z\lambda}$. □

Lemma 7. $\Pr[\mathbf{Non-abort}(\Gamma^{real})]$ *is negligibly close to* $\Pr[\mathbf{Non-abort}(\Gamma^{rand})]$.

Proof If the conclusion were not true, by using adversary $\mathcal{B}$, we show that there would exist j such that key assignment $C(I_j)$ does not satisfy key indistinguishability. We denote such an attacker by $\mathcal{O}$. He acts as follows.

1. $\mathcal{O}$ runs algorithm adversary $\mathcal{B}$ described in game Γ^{real}.
2. When $\mathcal{B}$ chooses j, $\mathcal{O}$ announces to have a test on S_j. As a response, he will receive all I_t for $I_t \in C(I_j)$ with $S_t \not\subseteq S_j$ as well as $\langle \alpha_{i_0}, \alpha_{i_1}, \cdots, \alpha_{i_h}\rangle$ taken from $\langle k_{i_0}, k_{i_1}, \cdots, k_{i_h}\rangle$ or $\langle r_0, r_1, \cdots, r_h\rangle$ uniformly random. Here r_t is uniformly random of length $|k_{i_t}|$ and k_{i_t} is the key associated with S_{i_t}, where $i_0 = j$ and $S_{i_1}, \cdots, S_{i_h}$ are all proper subsets of S_j with $I_{i_t} \in C(I_j), t = 1, \cdots, h$. Then $\mathcal{O}$ forwards all such information except α_{i_0} to adversary $\mathcal{B}$. Then $\mathcal{O}$ answers the encryption/decryption queries of $\mathcal{B}$ by using α_{i_0}.
3. If $\mathcal{B}$ does not abort, then $\mathcal{O}$ outputs 1 with probability $\frac{p_1}{p_1+p_2}$, where $p_1 = \Pr[\mathbf{Non-abort}(\Gamma^{rand})]$ and $p_2 = \Pr[\mathbf{Non-abort}(\Gamma^{real})]$. Otherwise, it outputs 1 with probability $\frac{p_2}{p_1+p_2}$.

Now we analyze the probabilities. Note that if $\langle \alpha_{i_0}, \alpha_{i_1}, \cdots, \alpha_{i_h}\rangle = \langle k_{i_0}, k_{i_1}, \cdots, k_{i_h}\rangle$, then the game initiated by $\mathcal{B}$ is exactly Γ^{real}. Thus the non-abort probability is exactly p_2. On the other hand, if $\langle \alpha_{i_0}, \cdots, \alpha_{i_h}\rangle = \langle r_0, \cdots, r_h\rangle$, the game initiated by $\mathcal{B}$ is distributed exactly the same as game Γ^{rand}. Let $\mathbf{Adv}(\mathcal{O})$ be the advantage of $\mathcal{O}$ in breaking the key indistinguishability of $C_1, \cdots, C_\mu$. Then we have

$$\begin{aligned}\mathbf{Adv}(\mathcal{O}) &= |\tfrac{1}{z}\textstyle\sum_{j=1}^{z}(\Pr[\mathcal{O}(r_0, \cdots, r_h, A_j) = 1 : j] - \Pr[\mathcal{O}(k_{i_0}, \cdots, k_{i_h}, A_j) = 1 : j])|\\ &= \left|\left(\tfrac{p_1}{p_1+p_2}p_1 + \tfrac{p_2}{p_1+p_2}(1-p_1)\right) - \left(\tfrac{p_1}{p_1+p_2}p_2 + \tfrac{p_2}{p_1+p_2}(1-p_2)\right)\right|\\ &= \tfrac{1}{p_1+p_2}|p_1^2 + p_2 - p_1p_2 - p_1p_2 - p_2 + p_2^2|\\ &= \tfrac{1}{p_1+p_2}(p_1 - p_2)^2\\ &> \tfrac{1}{2}(p_1 - p_2)^2.\end{aligned}$$

Since $p_1 - p_2$ is non-negligible, it follows that $\mathbf{Adv}(\mathcal{O})$ is non-negligible, a contradiction to Lemma 2. □

Lemma 8. *Suppose that key assignment on $C(I_j)$ for all j satisfies key indistinguishability, that F is semantically secure, and that E is pre-CCA secure. If $\mathcal{H}yb$ framework is insecure, then adversary $\mathcal{B}$ has a non-negligible advantage in game Γ^{real}.*

Proof We first prove that the output advantages in Γ^{real} and $\Gamma^{real'}$ are negligibly close. If this were not true, then there exists adversary $\mathcal{D}$ that can compromise the key assignment indistinguishability of $C(I_j)$ for some j. The action is similar to that in Lemma 6. The only difference is the output. In the special case, if $\mathcal{D}$ sent $E_{k'_{i_l}}(rd_{j_l})$ then he follows the output rule of $\mathcal{B}$ in game Γ^{real}; otherwise, he complements the output rule of $\mathcal{B}$ in game Γ^{real} (recall that in all variants of Γ^{real} $\mathcal{B}$ has the same output rule). Immediately, we have that the advantage $\mathbf{Adv}(\mathcal{D})$ of $\mathcal{D}$ is as follows.

$$\begin{aligned}\mathbf{Adv}(\mathcal{D}) &= \textstyle\sum_{l=1}^{\nu} \frac{\mathbf{Adv}(\Gamma_l^{real})-\mathbf{Adv}(\Gamma_{l-1}^{real})}{2\nu} \\ &= \frac{\mathbf{Adv}(\Gamma^{real})-\mathbf{Adv}(\Gamma^{real'})}{2\nu},\end{aligned}$$

non-negligible, contradiction.

Suppose a $\mathcal{H}yb$ scheme is insecure. Let $\mathcal{A}$ be the algorithm that is against $\mathcal{H}yb$ scheme. We can separate $\mathcal{A}$ as $(\mathcal{A}_1, \mathcal{A}_2)$. The job of $\mathcal{A}_1$ is to do the first part of the attack, which outputs (M, R) for test and as well as some auxiliary information α, where R contains all the users that are corrupted currently. And $\mathcal{A}_2$ is the second part of $\mathcal{A}$, which will receive the challenge ciphertext $\mathcal{H}(M', R)$ from the challenger and auxiliary information α from $\mathcal{A}_1$, where $M' = M$ or a random number of length $|M|$ equally likely. Then $\mathcal{A}_2$ outputs a guess bit for M'.

Define

$$\mathcal{H}_j(M,R) = \langle i_1, \cdots, i_m, E_{k_{i_1}}(r_1), \cdots, E_{k_{i_j}}(r_j), E_{k_{i_{j+1}}}(k), \cdots, E_{k_{i_m}}(k), F_k(M)\rangle \qquad (10)$$

to be a random variable over the distribution of R and its internal coins, where R is the output of $\mathcal{A}_1$. If $j > m$, let $\mathcal{H}_j(M,R) = \mathcal{H}_m(M,R)$.
Define

$$\epsilon_j = \Pr[\mathcal{A}_2(\mathcal{H}_j(M,R),\alpha) = 1 \text{ for } \alpha, M, R \leftarrow \mathcal{A}_1] - \Pr[\mathcal{A}_2(\mathcal{H}_j(M'',R),\alpha) = 1 \text{ for } \alpha, M, R \leftarrow \mathcal{A}_1],$$

where M'' is a random string of length $|M|, j = 0, \cdots, Q$. Here Q is an upperbound of m. Note that ϵ_0 is exactly the advatage of $\mathcal{A}$ in security definition of $\mathcal{H}yb$ scheme. Thus it is non-negligible according to the assumption. On the other hand, ϵ_Q is negligible by the semantic security of F and the fact that k happened to occur somewhere else during the attack only with negligible probability (since it is uniformly random).

Now let us analyze the advantage of $\mathcal{B}$ in game $\Gamma^{real'}$. For simplicity, we also separate $\mathcal{B}$ into two parts $(\mathcal{B}_1, \mathcal{B}_2)$. The job of $\mathcal{B}_1$ is to output k for test and some auxiliary information β. On receiving the challenge ciphertext $\gamma \in \{E_{k_j}(k), E_{k_j}(r_j)\}$ and β, the job of $\mathcal{B}_2$ is to output a guess bit.

From the proof of Lemma 6, we know that for given d, t, j, if $\mathcal{B}$ won't abort, then the view of adversary $\mathcal{A}$ in case "$\gamma \in E_{k_j}(k)$" is distributed exactly the same as in the real world conditional on the set of events $\Pi_{i_t=j}^{worl}(d)$ except that the challenge ciphertext $\mathcal{H}(M', R)$ is replaced by $\mathcal{H}_{t-1}(M', R)$. Note that $\mathcal{H}(M', R)$ is one-one correspondent to a set $\{\mathcal{H}_{t-1}(M', R)|r_1, \cdots, r_{t-1}\}$, where the random bits used in $\mathcal{H}_{t-1}(M', R)$ for given $r_1, \cdots r_{t-1}$ are the same as in $\mathcal{H}(M', R)$. Thus, for any such a view x in the real world, let $T_{t-1}(x)$ be the set of views in $\Gamma^{real'}$ that corresponds to x with parameter t such that $E_{k_{i_t}}(k)$ is contained in the challenge instead of $E_{k_{i_t}}(r_t)$. Then the probability that there exists an occurence of view in $T_{t-1}(x)$ conditional on fixed d, t, j and non-abortion event is $\frac{\Pr[x]}{\Pr[\Pi_{i_t=j}^{worl}(d)]}$. Note $\Pr[\Pi_{i_t=j}^{worl}(d)] = \sum_{x \in \Pi^{worl}(d)} \Pr[i_t = j, x]$.

Define $\Delta_1(t,j,d) = \Pr[\mathcal{B}_2(E_{k_j}(k), \beta) = 1 \text{ for } k, \beta \leftarrow \mathcal{B}_1 | d, t, j]$ to be the probability that $\mathcal{B}$ outputs bit 1 conditional on non-abortion event and fixed d, j, t in Step 1 of the game. Similarly, define $\Delta_2(t,j,d) = \Pr[\mathcal{B}_2(E_{k_j}(r_t), \beta) = 1 \text{ for } k, \beta \leftarrow \mathcal{B}_1 | d, t, j]$.

We have

$$\begin{aligned}\Delta_1(t,j,d) = &\textstyle\sum_{x\in \Pi^{worl}_{j=i_t}(d,0)} \frac{\Pr[x]}{\Pr[\Pi^{worl}_{i_t=j}(d)]} \Pr[\mathcal{A}_2(x' \in T_{t-1}(x)) = 1 : x] \\ &+ \textstyle\sum_{y\in \Pi^{worl}_{j=i_t}(d,1)} \frac{\Pr[y]}{\Pr[\Pi^{worl}_{i_t=j}(d)]} \Pr[\mathcal{A}_2(y' \in T_{t-1}(y)) = 0 : y],\end{aligned}$$

where $\Pi^{worl}_{i_t=j}(d,a)$ denotes the subset of $\Pi^{worl}_{i_t=j}(d)$ such that if M is used in the challenge ciphertext then $a = 0$; otherwise, $a = 1$.

Similarly,

$$\begin{aligned}\Delta_2(t,j,d) = &\textstyle\sum_{x\in \Pi^{worl}_{j=i_t}(d,0)} \frac{\Pr[x]}{\Pr[\Pi^{worl}_{i_t=j}(d)]} \Pr[\mathcal{A}_2(x' \in T_{t}(x)) = 1 : x] \\ &+ \textstyle\sum_{y\in \Pi^{worl}_{j=i_t}(d,1)} \frac{\Pr[y]}{\Pr[\Pi^{worl}_{i_t=j}(d)]} \Pr[\mathcal{A}_2(y' \in T_{t}(y)) = 0 : y].\end{aligned}$$

Notice that when $\mathcal{B}$ aborts, he will output 0 or 1 uniformly random. Thus the advantage of $\mathcal{B}$ comes from non-abort event only. Also notice that $\Pr[\Pi^{worl}_{i_t=j}(d,a)] = \Pr[\Pi^{worl}_{i_t=j}(d)]/2$. Thus the advantage $\mathbf{Adv}(\mathcal{B})$ of $\mathcal{B}$ is exactly the following

$$\mathbf{Adv}(\mathcal{B}) = \sum_{t,j,d} \Pr[t,j,d](\Delta_1(t,j,d) - \Delta_2(t,j,d)) \Pr[\Pi^{worl}_{i_t=j}(d)]/2. \tag{11}$$

We further have

$$\begin{aligned}\mathbf{Adv}(\mathcal{B}) = &\textstyle\sum_{j,t,d} \Pr[t,j,d] \sum_{x\in \Pi^{worl}_{j=i_t}(d,0)} \Pr[x] \Pr[\mathcal{A}_2(x' \in T_{t-1}(x)) = 1 : x]/2 \\ &+ \textstyle\sum_{t,j,d} \Pr[t,j,d] \sum_{y\in \Pi^{worl}_{j=i_t}(d,1)} \Pr[y] \Pr[\mathcal{A}_2(y' \in T_{t-1}(y)) = 0 : y]/2 \\ &- \textstyle\sum_{j,t,d} \Pr[t,j,d] \sum_{x\in \Pi^{worl}_{j=i_t}(d,0)} \Pr[x] \Pr[\mathcal{A}_2(x' \in T_{t}(x)) = 1 : x]/2 \\ &- \textstyle\sum_{t,j,d} \Pr[t,j,d] \sum_{y\in \Pi^{worl}_{j=i_t}(d,1)} \Pr[y] \Pr[\mathcal{A}_2(y' \in T_{t}(y)) = 0 : y]/2 \\ = &\textstyle\frac{1}{2Qz\lambda} \sum_{j,t,d} \sum_{x\in \Pi^{worl}_{j=i_t}(d,0)} \Pr[x] \Pr[\mathcal{A}_2(x' \in T_{t-1}(x)) = 1 : x] \\ &+ \textstyle\frac{1}{2Qz\lambda} \sum_{t,j,d} \sum_{y\in \Pi^{worl}_{j=i_t}(d,1)} \Pr[y] \Pr[\mathcal{A}_2(y' \in T_{t-1}(y)) = 0 : y] \\ &- \textstyle\frac{1}{2zQ\lambda} \sum_{j,t,d} \sum_{x\in \Pi^{worl}_{j=i_t}(d,0)} \Pr[x] \Pr[\mathcal{A}_2(x' \in T_{t}(x)) = 1 : x] \\ &- \textstyle\frac{1}{2zQ\lambda} \sum_{t,j,d} \sum_{y\in \Pi^{worl}_{j=i_t}(d,1)} \Pr[y] \Pr[\mathcal{A}_2(y' \in T_{t}(y)) = 0 : y]\end{aligned}$$

For a fixed t, any $x \in \Pi^{real}$ has a unique j such that $j = i_t$ (recall i_t is defined as i_m if $t > m$). Thus $\cup_j \Pi^{worl}_{j=i_t}(d,0)$ is the subset of $\Pi^{worl}(d)$ in which M is used in the challenge ciphertext for $\mathcal{A}$. Denote the union by $\Pi^{worl}(d,0)$. Furthermore, subsets in this union are pairwise disjoint. Similar observations are applied to other three cases. Thus we have

$$\begin{aligned}\mathbf{Adv}(\mathcal{B}) = &\textstyle\frac{1}{2Qz\lambda} \sum_{t,d} \sum_{x\in \Pi^{worl}(d,0)} \Pr[x] \Pr[\mathcal{A}_2(x' \in T_{t-1}(x)) = 1 : x] \\ &+ \textstyle\frac{1}{2Qz\lambda} \sum_{t,d} \sum_{y\in \Pi^{worl}(d,1)} \Pr[y] \Pr[\mathcal{A}_2(y' \in T_{t-1}(y)) = 0 : y] \\ &- \textstyle\frac{1}{2zQ\lambda} \sum_{t,d} \sum_{x\in \Pi^{worl}(d,0)} \Pr[x] \Pr[\mathcal{A}_2(x' \in T_{t}(x)) = 1 : x] \\ &- \textstyle\frac{1}{2zQ\lambda} \sum_{t,d} \sum_{y\in \Pi^{worl}(d,1)} \Pr[y] \Pr[\mathcal{A}_2(y' \in T_{t}(y)) = 0 : y]\end{aligned}$$

Further notice that any $x \in \Pi^{worl}$ has a unique D. Thus

$$\begin{aligned}\mathbf{Adv}(\mathcal{B}) &= \tfrac{1}{2Qz\lambda} \sum_t \sum_{x \in \Pi^{worl}(0)} \Pr[x] \Pr[\mathcal{A}_2(x' \in T_{t-1}(x)) = 1 : x] \\ &\quad + \tfrac{1}{2Qz\lambda} \sum_t \sum_{y \in \Pi^{worl}(1)} \Pr[y] \Pr[\mathcal{A}_2(y' \in T_{t-1}(y)) = 0 : y] \\ &\quad - \tfrac{1}{2zQ\lambda} \sum_t \sum_{x \in \Pi^{worl}(0)} \Pr[x] \Pr[\mathcal{A}_2(x' \in T_t(x)) = 1 : x] \\ &\quad - \tfrac{1}{2zQ\lambda} \sum_t \sum_{y \in \Pi^{worl}(1)} \Pr[y] \Pr[\mathcal{A}_2(y' \in T_t(y)) = 0 : y] \\ &= \tfrac{1}{2Qz\lambda} \sum_t \sum_{x \in \Pi^{worl}(0)} \Pr[x] \Pr[\mathcal{A}_2(x' \in T_{t-1}(x)) = 1 : x] \\ &\quad - \tfrac{1}{2Qz\lambda} \sum_t \sum_{y \in \Pi^{worl}(1)} \Pr[y] \Pr[\mathcal{A}_2(y' \in T_{t-1}(y)) = 1 : y] \\ &\quad - \tfrac{1}{2zQ\lambda} \sum_t \sum_{x \in \Pi^{worl}(0)} \Pr[x] \Pr[\mathcal{A}_2(x' \in T_t(x)) = 1 : x] \\ &\quad + \tfrac{1}{2zQ\lambda} \sum_t \sum_{y \in \Pi^{worl}(1)} \Pr[y] \Pr[\mathcal{A}_2(y' \in T_t(y)) = 1 : y],\end{aligned}$$

where $\Pi^{worl}(a) = \cup_d \Pi^{worl}(d, a)$ for $a \in \{0, 1\}$. Note that $\Pi^{worl}(0) \cap \Pi^{worl}(1) = \emptyset$ and $\Pi^{worl}(0)$ (resp. $\Pi^{worl}(1)$) is the subset of Π^{worl} such that M (resp. M'') is used in the challenge ciphertext. Therefore,

$$\begin{aligned}\mathbf{Adv}(\mathcal{B}) &= \tfrac{1}{2zQ\lambda} \sum_{t=1}^{Q} (\epsilon_{t-1} - \epsilon_t) \\ &= \tfrac{1}{2zQ\lambda} (\epsilon_0 - \epsilon_Q).\end{aligned}$$

Thus $\mathcal{B}$ has a non-negligible advantage. □

Lemma 9. *If adversary $\mathcal{B}$ in game Γ^{rand} has a negligible advantage while it has a non-negligible advantage in game Γ^{real}, then there exists an adversary $\mathcal{D}$ that compromises the key assignment indistinguishability of $\mathcal{H}yb$.*

Proof The action of $\mathcal{D}$ is the same as $\mathcal{O}$ in Lemma 7 except the output. Here $\mathcal{D}$ does the following:

1. If $\mathcal{B}$ aborts, then $\mathcal{D}$ outputs 0, 1 equally likely.
2. If $\mathcal{B}$ does not aborts and outputs 1, then $\mathcal{D}$ outputs 1 with probability $\frac{p_{real}}{p_{real}+p_{rand}}$; if $\mathcal{B}$ won't abort and outputs 0, then $\mathcal{D}$ outputs 1 with probability $\frac{p_{rand}}{p_{real}+p_{rand}}$, where p_{real} (resp. p_{rand}) is the probability $\mathcal{B}$ outputs 1 in game Γ^{real} (resp. Γ^{rand}) which is not due to abortion event.

For $ch \in \{real, rand\}$, let Π_0^{ch} denote the set of views of $\mathcal{A}$ in game Γ^{ch} with the abortion of $\mathcal{B}$ and Π_1^{ch} denote the set of views of $\mathcal{A}$ in game Γ^{ch} with non-abortion of $\mathcal{B}$. For simplicity, let $\mathcal{D} = (\mathcal{D}_1, \mathcal{D}_2)$, where the job of $\mathcal{D}_1$ is to output j and the job of $\mathcal{D}_2$ is to do the rest job. Then we have

$$\begin{aligned}
\mathbf{Adv}(\mathcal{D}) &= |\Pr[\mathcal{D}_2(r_0, \cdots, r_h, A_j) = 1 \text{ for } j \leftarrow \mathcal{D}_1] - \Pr[\mathcal{D}_2(k_{i_0}, \cdots, k_{i_h}, A_j) = 1 \text{ for } j \leftarrow \mathcal{D}_1]| \\
&= |\Pr[\mathcal{B}(x \in \Pi_1^{real}) = 1] \cdot \tfrac{p_{real}}{p_{real}+p_{rand}} + (1 - \Pr[\mathbf{Non-abort}(\Gamma^{real}))/2 \\
&\quad -(\Pr[\mathcal{B}(x \in \Pi_1^{rand}) = 1] \cdot \tfrac{p_{real}}{p_{real}+p_{rand}} + (1 - \Pr[\mathbf{Non-abort}(\Gamma^{rand})])/2) \\
&\quad + \Pr[\mathcal{B}(x \in \Pi_1^{real}) = 0] \cdot \tfrac{p_{rand}}{p_{real}+p_{rand}} - \Pr[\mathcal{B}(x \in \Pi_1^{rand}) = 0] \cdot \tfrac{p_{rand}}{p_{real}+p_{rand}}| \\
&\approx |(\Pr[\mathcal{B}(x \in \Pi_1^{real}) = 1] - \Pr[\mathcal{B}(x \in \Pi_1^{rand}) = 1]) \cdot \tfrac{p_{real}-p_{rand}}{p_{real}+p_{rand}}| \\
&= \tfrac{(p_{real}-p_{rand})^2}{p_{real}+p_{rand}} \\
&= \tfrac{(\mathbf{Adv}^{real}(\mathcal{B})-\mathbf{Adv}^{rand}(\mathcal{B}))^2}{4(p_{real}+p_{rand})}, \\
&\geq \tfrac{(\mathbf{Adv}^{real}(\mathcal{B})-\mathbf{Adv}^{rand}(\mathcal{B}))^2}{8},
\end{aligned}$$

where $\approx$ means "negligibly close" and $\mathbf{Adv}^{ch}(\mathcal{B})$ is the advantage of $\mathcal{B}$ in game Γ^{ch}, for $ch \in \{real, rand\}$. Therefore, $\mathbf{Adv}(\mathcal{D})$ is non-negligible. □

Proof of Theorem 1 The theorem directly follows from Lemmas 7, 8 and 9. □