Multi-round Secure Light-Weight Broadcast Exclusion Protocol with Pre-processing

Yuji Watanabe and Masayuki Numao

IBM Research, Tokyo Research Laboratory
1623-14 Shimotsuruma, Yamato-shi,
Kanagawa, 242-8502, Japan
{muew,numao}@jp.ibm.com

Abstract. A broadcast exclusion protocol allows a broadcaster to transmit a encrypted message to a set of n users over a broadcast channel so that all but some specified small group of k excluded users can decrypt the message, even if these excluded users collude with each other in an arbitrary manner. Recently, Matsuzaki et al. pointed out a potential problem in the earlier works regarding the number of modular exponentiation, and proposed an extended scheme in which decryption requires only two modular exponentiations regardless of n and k. However, our analysis shows this scheme has a limitation of the number of rounds.
The contribution of this paper is to present a new broadcast exclusion protocol maintaining security within a virtually unlimited number of rounds without spoiling the efficiency. First, we demonstrate a limitation of the rounds of the previous work by showing how a user can derive the system secret parameters after more than a certain number of rounds. Then, we present a new protocol for which we can provide rigorous security proof under the Computational Diffie-Hellman (CDH) assumption.
We note that even if we point out some limitation of the previous work, we still consider it nevertheless significant. In particular, we derived our new protocol by modifying some of their fundamental techniques.

Keywords: broadcast encryption, broadcast exclusion problem, pre-processing

1 Introduction

1.1 Background

Consider a situation where a large amount of data is distributed to all authorized users over a broadcast channel. Typically, a broadcaster provides each authorized user with a private decryption key, then the data is broadcasted in an encrypted form. Finally, the authorized users are able to decrypt the data and obtain the service they intended to get. This scenario can come up in the context of Cable, pay-TV, Internet multicasts, satellite, and group telecommunications.

In the above situation, it is required for the broadcaster to share a key (which is called the *broadcast key* in this paper) with all the non-excluded users quickly

E. Snekkenes and D. Gollmann (Eds.): ESORICS 2003, LNCS 2808, pp. 85–99, 2003.

and securely. A broadcast encryption, introduced by Fiat and Naor [1], allows a broadcaster to distribute the broadcast key simultaneously and securely over a broadcast channel to selected subsets of users. A number of derivative works have been done mainly toward minimizing the communication overhead and the storage requirements for each user.

The problem of transmitting a message only to some small specified subset of the users has been considered in [1]. On the other hand, the broadcast exclusion problem, which is a kind of broadcast encryption, is the problem of transmitting a message over a broadcast channel shared by a number n of users so that all but some specified small coalition of k excluded users can decrypt the message, even if these excluded users collude with each other in an arbitrary manner. This paper addresses the broadcast exclusion protocol, in which a broadcast key is transmitted to each user in an encrypted form (we call this ciphertext "*header*") over a broadcast channel so that all but some specified subset of users can get the broadcast key[1].

If the private key of a user is exposed, for example, the user must be excluded as quickly as possible in order to prevent an eavesdropper from accessing the broadcasted secret information. The broadcast exclusion protocol allows the broadcaster to exclude the user from the recipients. Trivially, the broadcast exclusion protocol can be realized by a broadcaster sending a new broadcast key to each user except for the excluded one. This is optimal regarding the user's storage size, while the communication overhead is $O(n)$, and thus it takes a long time to send the broadcast key if n is large.

Several results on the broadcast exclusion problem have been proposed so for. Kumar, Rajagopalan and Sahai [4] proposed a broadcast exclusion protocol based on an error correcting code without any computational assumption. In their scheme, the communication overhead is $O(k^2)$ regardless of n, while the size of the private keys stored by each user is $O(k \log n)$, still depending on n.

Another method proposed by Anzai, Matsuzaki and Matsumoto [5] employs threshold cryptosystems [6] in order to avoid any dependence on n for the size of the private keys. A similar result was discovered independently by Naor and Pinkas [7] (In addition, their work includes the tracing and self enforcement extensions). The underlying idea of their schemes is as follows. A broadcaster divides a secret s into $n+k$ shares with a technique for a $(k+1, n+k)$-threshold scheme, and distributes a shares to each of the n users in a secure way. When the broadcaster excludes d users, k shares including d shares of the excluded users are broadcasted to all of the users. A non-excluded user can recover the broadcast key using $k+1$ shares, i.e., his share and k broadcasted shares, while the excluded users cannot do so, since the broadcasted shares contain his own share, and thus he can only obtain k shares. In their schemes, the communication overhead is $O(k)$ regardless of n, and the size of the private key that each user stores is $O(1)$. Their schemes also work in a public-key setting in the sense that there is no requirement for a fixed-privileged broadcaster. Moreover, the schemes are used

[1] Our protocol assumes the user is *stateful* in the sense that the users update their state from round to round. The stateless cases have been studied in [2], [3] etc.

multiple rounds, i.e., a single initialization enables multiple rounds of execution of broadcast exclusion among a set of users. Their schemes, however, require each user to compute the broadcast key with $O(k)$ modular exponentiations. The parameter k should be enough large to withstand the collusion, while setting a large value for k results in computing a large number of modular exponentiations, which takes a long time especially when the user's computing device has low computational power, such as a pervasive device.

The recent result in [8] (by the same authors as [5]) proposed an innovative technique to extend the scheme of [5] in such a way that each user computes the broadcast key with just two modular exponentiations, regardless of n and k. The transmission overhead is $O(k)$ and the size of the private key each user stores is $O(1)$, thus there are no changes in those orders from [5]. As an additional assumption, this scheme requires a fixed-privileged broadcaster which knows all the secret parameters and determines the excluded users. This is considered to be reasonable in some actual applications.

However, our analysis shows this scheme [8] has a limitation of the number of rounds. Actually, a user who observes the communications from the broadcaster for more than a certain number of rounds can calculate the secret parameters and defeat the subsequent exclusion. It is desirable to avoid such limitation of the number of rounds without spoiling the efficiency of computing the broadcast key.

1.2 Our Contribution

Computing modular exponentiation comprises the major portion of the computation for updating the broadcast key. For example, modular exponentiation requires t squarings and $t/2$ multiplications by using the general square-and-multiply algorithm, where the exponent can be expressed by t bits. Thus, reducing the number of modular exponentiations makes a substantial contribution to lowering the workload of broadcast key computations, which allows for the quick initiation of a new round.

It is required in [5] to compute the product of $k+1$ exponentials with distinct bases and distinct exponents. The authors of [7] mentioned that in this situation, a simultaneous multiple exponentiations algorithm can be used to reduce the computation overhead, where it takes k squarings and $(2^{k+1} - 2) + t - 1$ multiplications with $O(2^k)$ pre-computed values. However, this is effective only for small k. If k is large, e.g., $k = 1000$, the pre-computed values and run-time multiplications cannot be processed in a practical timeframe.

The computational overhead to derive the broadcast key is divided into two parts, the pre-processing and the real-time processing, which means a part of the computation before and after receiving the header, respectively.. The pre-processing can be performed during the previous round, while the real-time processing should be done within a short period of time just after receiving the header. Thus, the real-time processing would be the most significant part for rapid initiation of the new round. Meanwhile, we assume that it is not significant if pre-processing takes a long time to compute. This is reasonable for many

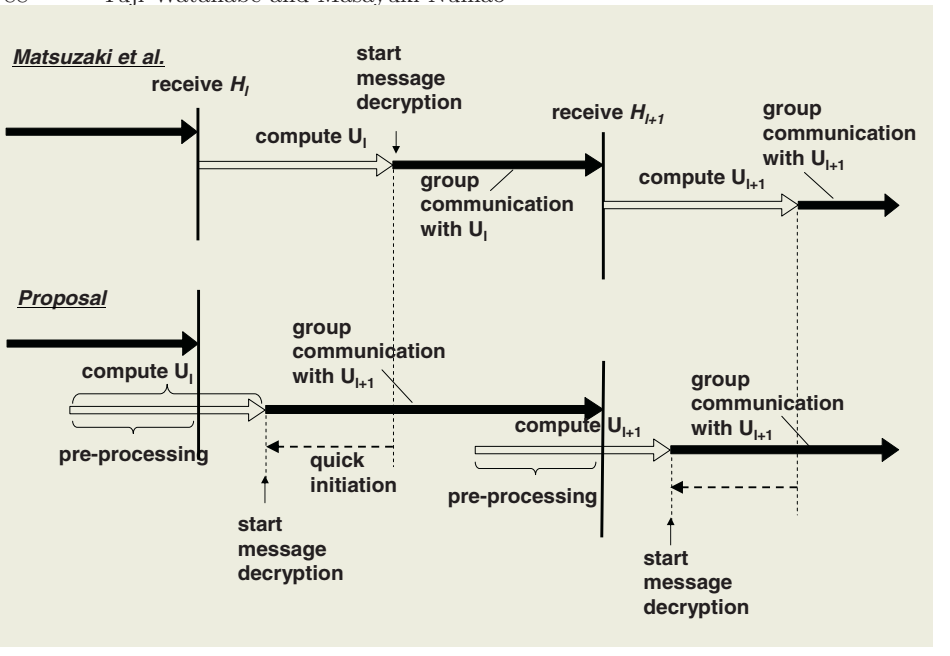

Fig. 1. Rapid decryption of new broadcast key using pre-processing

applications unless changes of the rounds happen frequently. As shown in Figure 1, we use the fact that shifting a part of the computation from the real-time processing into the pre-processing makes a significant improvement to accelerate the initiation of the rounds even if the overall computational overhead remains unchanged.

Our contribution provides a new efficient broadcast exclusion protocol maintaining the security for a virtually unlimited number of rounds while restraining the number of modular exponentiations in the real-time processing $O(1)$. First, we demonstrate a limitation of the rounds in [8] by showing how a user who observes the communications from the broadcaster for more than a certain number of rounds can derive the system secret parameters, which implies a total break of the scheme. Then, we present a new efficient broadcast exclusion protocol for which we provide a rigorous security proof by exploiting a strategy similar to that presented in [9][10]. Therefore, our protocol is proven secure against the collusion among up to k excluded users under the Computational Diffie-Hellman (CDH) assumption.

Table 1 shows a comparison of [5], [8], and ours. Apparently, [5] requires $O(k)$ modular exponentiations during the real-time processing for computing the new broadcast key, and there exists no trivial way to shift the computation to the pre-processing. The approach of [8] and our proposal require only two modular exponentiations during the real-time processing, but only ours has no limitation on the rounds. Thus, as far as we know, this result is the first one which satisfies the above properties. Another difference between [5] and our protocol

Table 1. Comparison

Description	[5]	[8]	proposal
num. of exp. (real-time processing)	$O(k)$	$O(1)$	$O(1)$
num. of exp. (pre-processing)	0	0	$O(k)$
size of header	$O(k)$	$O(k)$	$O(k)$
size of private key	$O(1)$	$O(1)$	$O(1)$
message distributor	arbitrary	$\mathcal{B}$ only	$\mathcal{B}$ only
num. of rounds	unlimited	limited	unlimited

is to require a fixed-privileged broadcaster. As with [8], this is considered to be reasonable in many applications.

We note that even if we point out some limitations of [8], we consider their work nevertheless significant. In particular, they found a potential problem in the earlier work and introduced important steps to reduce the amount of computation. We use some of their fundamental techniques as basic building blocks.

1.3 Organization

Our paper is structured as follows: In Section 2, we describe the model of our broadcast exclusion protocol and the definitions used in this paper. Section 3 shows some limitations on the multi-round security of the preceding works. We present our new protocol in Section 4, and show the security proof of the protocol in Section 5. Finally, we shall conclude in Section 6 with a summary of our results and issues that still remain to be solved.

2 Preliminaries

2.1 Definitions

We define the following notations used throughout the paper. Suppose a broadcaster, denoted as $\mathcal{B}$, wants to set up a broadcast communication channel only to a set of non-excluded users so that the set of excluded users chosen by $\mathcal{B}$ cannot get any information from the channel within a certain period of time, which is referred to as a round. $\mathcal{B}$ wants to be able to flexibly choose such a set of excluded users for every round. Let $\Phi = \{1, \ldots, n\}$ be the set of all users in the whole system, and $n = |\Phi|$ be the number of users.

At the time of system setup, $\mathcal{B}$ determines the system parameters and distributes a private key key_v to each user v in Φ. For the l-th round, the users in Φ are classified into a set of excluded users and a set of authorized (or non-excluded) users, denoted as Λ_l and Ω_l, respectively. We say a user v is authorized in the l-th round if v can compute a key U_l shared among $\mathcal{B}$ and Φ (which is called the broadcast key in this paper) from the header H_l using its private key key_v. We also say a user v' is excluded in the l-th round if v' cannot compute U_l from H_l. For simplicity, $\Omega_0 = \Phi$ and $\Lambda_0 = \{\emptyset\}$. The model of broadcast exclusion protocol we focus on in this paper is described as follows.

1. First, $\mathcal{B}$ sends to a user v ($\in \Phi$) the initial broadcast key U_0 and v's private key key_v over a secure private channel. $\mathcal{B}$'s public key is broadcasted to Φ.
2. The following procedures are iterated for $l \geq 1$ (we call the l-th iteration the "l-th round".)
   (a) $\mathcal{B}$ determines Λ_l. (i.e., $\Omega_l = \Phi \setminus \Lambda_l$).
   (b) $\mathcal{B}$ computes the header, denoted as H_l, and broadcasts it to Φ.
   (c) $v \in \Omega_l$ can compute U_l with H_l and key_v, while $v' \notin \Omega_l$ cannot do so.
   (d) U_l is used for $\mathcal{B}$ to securely send any message to Ω_l over a broadcast channel.

Remarks: The rule of exclusion is different between [8] and our protocol. In the scheme in [8], the user that is excluded once can never perform subsequent decryption of updated broadcast keys due to security reason, i.e., $\Lambda_l \supset \Lambda_{l-1}$. This means users excluded once have their permission permanently revoked unless their private keys are retransmitted[2]. On the other hand, our protocol allows a set of excluded users to be chosen independently for every round, i.e., the excluded user can continue to decrypt headers for the subsequent rounds without resending the new private key to him. This property will be nice when a subscriber does not pay a fee for a daily charged service, and thus the service must be stopped until the fee is paid. If he paid the fee, he can continue the decryption without any private communication with the broadcaster.

In another cases, however, it might be desirable that an excluded user stays excluded and can restart decryption only if the broadcaster resends him another set of keys. Our protocol can be modified in such a way that the excluded user cannot decrypt the headers for the further rounds only by encrypting the header information with the previous round broadcast key. In practice, the broadcast key might be distributed in the above two ways simultaneously to handle fine-grained control of the exclusion. For example, the former will be used for temporary exclusion at short interval, e.g., once a day, while the latter will be used for permanent exclusion at longer interval, e.g., once a week.

The parameters used in this paper are defined here. Let p be a prime power, q be a prime such that $q|p-1$, and g be a q-th root of unity over $GF(p)$ and $\langle g \rangle$ be a subgroup in $GF(p)$ generated by g. All the participants agree on p, q, g and $\langle g \rangle$. All arithmetic operations in this article are done in $GF(p)$ hereafter unless otherwise noted.

We assume that no polynomial-time algorithm solves $\log_g h$ in Z_q only with negligible probability in the size of q when h is selected uniformly from $\langle g \rangle$. We also make stronger assumptions: the intractability of the Computational Diffie-Hellman problem (CDH) in $\langle g \rangle$. We say that CDH in the group $\langle g \rangle$ is intractable if no probabilistic polynomial-time algorithm can find $g^{uv} \in \langle g \rangle$ from g^u and g^v.

Let $E_K(m)$ be a ciphertext given by symmetric key encryption of m with the key K. The discussion in this paper assumes that the underlying symmetric encryption $E()$ is sufficiently secure. Let k be the maximum number of excluded

[2] To support this feature, the header H_l is an encrypted message with the previous broadcast key U_{l-1}. See Section 3.

users for a single round. In our model of broadcast exclusion, we assumes $k \ll n$, e.g., k=1,000 and n=1,000,000.

2.2 Requirements

This paper presents a solution to meet the following requirements for the security and the efficiency constraints.

1. The user $v \in \Omega_l$ which is authorized in the l-th round can compute the broadcast key U_l (within a deterministic polynomial time in the size of $\langle g \rangle$).
2. An adversary who controls at most k users excluded in the l-th round cannot compute the broadcast key U_l (within a probabilistic polynomial time in the size of $\langle g \rangle$).W
3. The length of the header H_l and the size of the private key key_v do not depend on the number of users n.
4. The number of modular exponentiations in the *real-time processing* does not depend on either n or k, where "real-time processing" means the part of the computation which must be performed in order to derive U_l after receiving H_l.
5. The protocol does not have any limitation of the number of rounds.

In Requirement 2, we regard the adversary as an algorithm that can access all the information that can be accessed by up to k excluded users before the attack. More precisely, the adversary $\mathcal{A}_R$ who controls k excluded users in Λ_R can access the initial broadcast key U_0, the private keys of the k excluded users, the header $H_1, \ldots, H_R$, and the public information, where we say the protocol meets Requirement 2 if $\mathcal{A}_R$ cannot compute U_R within a probabilistic polynomial time in the size of $\langle g \rangle$. A more rigorous definition appears in Section 5.

2.3 Related Works

We briefly overview earlier work that focuses on the issues described above. The schemes in [4] and [2] presented solutions to Requirements 1 and 2 without making any computational assumptions, while the length of the header and the size of the private keys depend on the number n, and therefore Requirement 3 still remains to be solved. The schemes in [5] and [8] meet Requirement 3, since the length of the header is $O(k)$ and the size of the private key is $O(1)$. The scheme in [5] requires $O(k)$ modular exponentiations for each user to compute the updated broadcast key, while the scheme in [8] requires only two modular exponentiations, which implies [8] also meets Requirement 4.

However our analysis described in the following section shows that [8] has a limitation of the number of rounds. More precisely, more than a certain number of updates of the broadcast key results in a total break of the system. An objective of this paper is to modify the protocol of [8] to make it a multi-round secure one without spoiling the important feature of eliminating the $O(k)$ modular exponentiations. Our protocol is closely based on their original work in [5] and [8]. Our protocol actually uses some of the techniques suggested there, resulting in a protocol which can be proven secure under the CDH-assumption.

3 Multi-round Security of the Light-Weight Broadcast Exclusion Protocol

Before the detailed discussion of the security in [8], we present an outline of that protocol below.

1. (Setup) $\mathcal{B}$ determines the maximum number of excluded users, denoted as k, and randomly chooses two univariate polynomials of degree k over Z_q, denoted as $F(x) = \sum_{j=0}^{k} a_j x^j$, $G(x) = \sum_{j=0}^{k} b_j x^j$, where $F(0) = S$ and $G(0) = T \pmod q$ are secret values only $\mathcal{B}$ knows. Then, $\mathcal{B}$ chooses the initial broadcast key U_0 from $\langle g \rangle$ at random, and sends a pair consisting of U_0 and the private key $key_i = (s_i, f_i) = (F(i), g^{G(i)/F(i)})$ to each user i in Φ over a secure private channel.
2. (Encryption of the broadcast key) Transmitting the l-th round broadcast key U_l is done as follows. First, $\mathcal{B}$ randomly chooses r_l from Z_q, and computes $X_l = g^{r_l}$. Then it determines a set Δ_l of d users to be excluded from Ω_l. In the scheme in [8], any user that is excluded once can never perform subsequent decryption of updated broadcast keys. Thus, excluding users once means their permissions are permanently revoked unless their private keys are retransmitted (formally, $\Lambda_l = \Lambda_{l-1} \cup \Delta_l$ and $\Omega_l = \Omega_{l-1} \setminus \Delta_l$). $\mathcal{B}$ selects a set Θ_l of $k - d$ integers from $Z_q \setminus (\Omega_l \cup (\bigcup_{j=1}^{l-1} \Theta_j))$, and then computes M_{lj} $(j \in \Delta_l \cup \Theta_l)$ as
$$M_{lj} = r_l F(j) + G(j) \bmod q \quad .$$
Finally, $\mathcal{B}$ computes the header H_l as
$$H_l = E_{U_{l-1}}(B_l) = E_{U_{l-1}}(X_l \| \{(j, M_{lj}) \mid j \in \Delta_l \cup \Theta_l\})$$
and broadcasts it to every user, where U_{l-1} is the $(l-1)$-th broadcast key, and $\mathcal{B}$ can compute a new broadcast key U_l shared among all of the authorized users in the l-th round as $U_l = g^{r_l S + T}$.
3. (Decryption of the broadcast key) The user $v \in \Omega_l$ authorized in the l-th round has already obtained U_{l-1} before the $(l-1)$-th round because if $v \in \Omega_l$, then $v \in \Omega_{l-1}$. Then v computes B_l by decrypting the received ciphertext $E_{U_{l-1}}(B_l)$ with U_{l-1}, and obtains the new broadcast key U_l as
$$U_l = (X_l f_v)^{W_{l1}} g^{W_{l2}}$$
where W_{l1} and W_{l2} are represented by
$$W_{l1} = s_v L(v) \bmod q$$
$$W_{l2} = \sum_{j \in \Delta_l \cup \Theta_l} (M_{lj} L(j)) \bmod q$$
where $L(j)$ is a Lagrange interpolation coefficient which can be derived from
$$L(j) = \prod_{t \in \Delta_l \cup \Theta_l \cup \{v\} \setminus \{j\}} t/(t - j) \bmod q \ .$$

In the above schemes, the number of modular exponentiations required to compute U_l is only two regardless of n and k. The scheme, however, has a limitation of the number of rounds. Consider any non-excluded user in the R-th round, denoted as $v \in \Omega_R$, who obtains (j, M_{lj}) $(j \in \Delta_l \cup \Theta_l)$ in the l-th round for $l = 1, \ldots, R$, where the following equations are satisfied.

$$M_{lj} = r_l \sum_{t=0}^{k} a_t j^t + \sum_{t=0}^{k} b_t j^t \bmod q$$

$$(l = 1, \ldots, R,\ j \in \Delta_l \cup \Theta_l,\ |\Delta_l \cup \Theta_l| = k)$$

The value of j is known, while the values of $a_0, \ldots, a_k$, $b_1, \ldots, b_k$, and r_l are unknown. Thus, these equations gives a system of kR equations w.r.t. $2k+2+R$ variables, $a_0, \ldots, a_k$, $b_0, \ldots, b_k$, $r_1, \ldots, r_R$. According to this reduction, if there is an algorithm to solve the system of equations, v can compute all of those $2k + 2 + R$ variables even if the discrete logarithm problem is hard to compute. We remark that $S(= a_0)$ and $T(= b_0)$ are the secret information of $\mathcal{B}$, so the exposure of these secrets implies a total break of the system.

There is a chance to solve systems of equations if the number of equations exceeds the number of variables. This happens if $kR \geq 2k+2+R$, i.e., $R \geq 3$ for any k (≥ 5). However, in general, solving large systems of quadratic multivariate polynomial equations is NP-hard over any field. When the number of equations γ_m is the same as the number of unknowns γ_n, the best known algorithms are exhaustive search for small fields, and a Gröbner base algorithm for large fields[11][12]. Gröbner base algorithms have large exponential complexity and cannot solve large systems in practice.

The number of equations becomes larger than the number of equations after a certain number of rounds. Recently, efficient algorithms for solving overdefined systems of multivariate polynomial equations have been proposed in [13]. The asymptotic complexity of their algorithm is expected to be polynomial with an exponent of $O(1/\sqrt{\epsilon})$ if the number of equations γ_m and the number of variables γ_n are related by $\gamma_m \geq \epsilon {\gamma_n}^2$ for any constant $0 < \epsilon \leq 1/2$ and the maximum degree D approximates $\lceil 1/\sqrt{\epsilon} \rceil$.

As the rounds proceed, the number of equations and the number of variables are increased simultaneously, whereas it is easy to see that solving the above system of kR equations of k degree with $2k + 2 + R$ variables can be reduced into solving a system of $(k-1)R$ quadratic equations with $2k + 2$ variables (fixed regardless of R), as follows: First, the equations in the l-th round can be represented by

$$\begin{pmatrix} M_{l1} \\ M_{l2} \\ \vdots \\ M_{lk} \end{pmatrix} = \begin{pmatrix} r_l a_0 + b_0 \\ r_l a_0 + b_0 \\ \vdots \\ r_l a_0 + b_0 \end{pmatrix} + B \begin{pmatrix} r_l a_1 + b_1 \\ r_l a_2 + b_2 \\ \vdots \\ r_l a_k + b_k \end{pmatrix} \quad ,$$

where B is a $k \times k$ matrix.

$$B = \begin{pmatrix} j_{l1} & {j_{l1}}^2 & \cdots & {j_{l1}}^k \\ j_{l2} & {j_{l2}}^2 & \cdots & {j_{l2}}^k \\ \vdots & \vdots & & \vdots \\ j_{lk} & {j_{lk}}^2 & \cdots & {j_{lk}}^k \end{pmatrix}$$

B is nonsingular because it is a Vandermonde matrix. Therefore,

$$\begin{pmatrix} r_l a_1 + b_1 \\ r_l a_2 + b_2 \\ \vdots \\ r_l a_k + b_k \end{pmatrix} = B^{-1} \begin{pmatrix} M_{l1} - r_l a_0 - b_0 \\ M_{l2} - r_l a_0 - b_0 \\ \vdots \\ M_{lk} - r_l a_0 - b_0 \end{pmatrix} \quad ,$$

Each line consists of a linear equation with respect to r, and thus k equations can be reduced to $k-1$ equations that have only $2k+2$ variables, $a_0, \ldots, a_k$ and $b_0, \ldots, b_k$. As a result, the kR equations with $2k+2+R$ variables can be reduced to $(k-1)R$ equations with $2k+2$ variables. The solving algorithm in [13] works efficiently if

$$(k-1)R \geq \epsilon(2k+2)^2 \;\Leftrightarrow\; R \geq \frac{\epsilon(2k+2)^2}{k-1} \quad \approx 4\epsilon k$$

for any constant $0 < \epsilon \leq 1/2$. This implies that the scheme in [8] has a limitation regarding the number of rounds.

4 Protocol

The goal of this paper is to provide the two desirable properties: (1) rapid computation of broadcast keys, and (2) no limitation of the number of rounds. The main idea is to reduce the number of exponentiations in the *real-time processing* by employing a technique for pre-processing. The underlying mechanism is based on the schemes in [5] and [8]. Our protocol is described as follows:

Setup. $\mathcal{B}$ determines the maximum number of excluded users, denoted as k, and randomly chooses two univariate polynomials of degree k over Z_q, denoted as $F(x) = S + \sum_{j=1}^{k} a_j x^j$, $G_1(x) = T_1 + \sum_{j=1}^{k} b_{1j} x^j$, where $F(0) = S$ and $G(0) = T_1 \pmod q$ are secret values only $\mathcal{B}$ knows. Then $\mathcal{B}$ chooses the initial broadcast key U_0 from $\langle g \rangle$ at random, and sends a pair consisting of U_0 and the private key of the first round, denoted as $key_i = (s_i, f_{1i}) = (F(i), g^{G_1(i)/F(i)})$, to each user i in Φ over a secure private channel.

Encryption of the broadcast key for the *l*-th round. Transmitting the l-th round broadcast key U_l is done as follows. First, $\mathcal{B}$ randomly chooses r_l from Z_q, and computes $X_l = g^{r_l}$. Then it determines a set Λ_l of d users to be excluded. In our protocol, a set of excluded users are independently chosen for each round, which is a different feature from the protocol in [8]. Thus,

excluding users in a round means they are revoked only until the broadcast key is refreshed, i.e., $\Lambda_l \subset \Phi$ and $\Omega_l = \Phi \setminus \Lambda_l$. $\mathcal{B}$ selects a set Θ_l of $k-d$ integers from $Z_q\Phi$, and then computes M_{lj} $(j \in \Lambda_l \cup \Theta_l)$ as

$$M_{lj} = r_l F(j) + G_l(j) \bmod q \ ,$$

and constructs B_l, the information for a user in Ω_l to compute U_l, as

$$B_l = \langle X_l \parallel \{(j, M_{lj}) | j \in \Lambda_l \cup \Theta_l\} \rangle \ ,$$

where $\mathcal{B}$ can compute U_l by $U_l = g^{r_l S + T_l}$. Then, $\mathcal{B}$ produces the information C_l for pre-processing a part of the computation of the next round broadcast key U_{l+1}. $\mathcal{B}$ chooses $b_{l+1,j}$ $(j = 0, \ldots, k)$ from Z_q at random, and computes $(u_{lj} = g^{b_{l+1,j}}$, where C_l can be represented by

$$C_l = (u_{l0} \| \ldots \| u_{lk}) \quad .$$

Finally, $\mathcal{B}$ computes the header H_l, which can be derived from

$$H_l = (B_l \| C_l) \quad ,$$

and broadcasts it to every user.

Decryption of the broadcast key for the *l*-th round. The user v receives H_l, which contains B_l and C_l, and checks if v is authorized in this round. If so, v can compute U_l using

$$U = (X_l f_{lv})^{W_{l1}} g^{W_{l2}}$$

$$W_{l1} = s_v L(v) \bmod q$$

$$W_{l2} = \sum_{j \in \Lambda \cup \Theta} M_{lj} L(j) \bmod q \ ,$$

where

$$L(j) = \prod_{t \in \Lambda \cup \Theta \cup \{v\} \setminus \{j\}} t/(t-j) \bmod q \ .$$

Pre-processing for computing U_l. Before receiving H_{l+1}, each user v performs pre-processing with

$$f_{l+1,v} := \left(\prod_{j=0}^{k} {u_{lj}}^{v^j} \right)^{1/s_v}$$

For computing H_{l+1}, $\mathcal{B}$ updates T_{l+1} and $G_{l+1}(x)$ as

$$T_{l+1} := b_{l+1,0} \bmod q$$

$$G_{l+1}(x) := \sum_{j=0}^{k} b_{l+1,j} x^j \bmod q.$$

5 Security

Suppose an adversary $\mathcal{A}_R$ who controls k users excluded in the R-th round, denoted as Λ_R, tries to find the R-th broadcast key U_R. As mentioned in Requirement 2, $\mathcal{A}_R$ can be formulated as a probabilistic algorithm $\mathcal{M}_1$ which given all the information observed by Λ_R outputs U_l.

Definition 1 (Adversary). *$\mathcal{M}_1$ is a probabilistic polynomial time algorithm in the size of $\langle g \rangle$, which takes the following values and outputs U_R with a negligible probability.*

- *public information : $g.p, q, k$*
- *private keys of k excluded users in Λ_R : $\{(s_j, f_{1j}) \mid j \in \Lambda_R\}$*
- *the initial broadcast key : U_0*
- *the header : $H_1, \ldots, H_R$*

Theorem 1. *$\mathcal{M}_1$ is as hard as CDH in the group $\langle g \rangle$.*

Proof. (Sketch)
Let $\mathcal{M}_2$ be the probabilistic polynomial time algorithm to solve CDH.
$(\mathcal{M}_2 \rightarrow \mathcal{M}_1)$ It is clear that the existence of $\mathcal{M}_2$ implies the existence of $\mathcal{M}_1$.
$(\mathcal{M}_1 \rightarrow \mathcal{M}_2)$ Suppose there exists $\mathcal{M}_1$. We show $\mathcal{M}_2$ by using $\mathcal{M}_1$ as a subroutine. Let the input to $\mathcal{M}_2$ be (α, β). The goal is to compute $\gamma = \beta^{\log_g \alpha}$ by using $\mathcal{M}_1$ as a subroutine. Intuitively, the input to $\mathcal{M}_1$ is constructed by using the following strategy. In the setup phase, $\mathcal{M}_2$ determines $F(x)$ such that $F(0) = S = \log_g \alpha$, and then computes all the values given to $\mathcal{M}_1$ with T_R and X_R such that T_R is chosen at random from Z_q and $r_R = \log_g \beta$. After the setup, $\mathcal{M}_2$ invokes $\mathcal{M}_1$ and obtains its output $U_R = g^{r_R S + T_R}$. Finally, $\mathcal{M}_2$ computes $g^{r_R S} = U_R / g^{T_R}$, which implies $\mathcal{M}_2$ can derive $\gamma = \beta^{\log_g \alpha}$. The procedure using M_2 proceeds as follows. Let a set of k integers in $\Lambda_l \cap \Theta_l$ be $\Gamma_l = \{\psi_{l1}, \ldots, \psi_{lk}\}$.

1. R set of k users $\Gamma_1, \ldots, \Gamma_R$ are uniformly chosen from Φ, where $\Lambda_R = \Gamma_R$.
2. The following parameters are uniformly chosen from Z_q.
   - s_j $(j \in \Lambda_R)$
   - r_l $(l = 1, \ldots, R-1)$
   - T_l $(l = 1, \ldots, R)$
   - M_{lj} $(l = 1, \ldots, R,\ j \in \Gamma_l)$
   - $b_{R+1,t}$ $(t = 0, \ldots, k)$
3. U_0 is uniformly chosen from $\langle g \rangle$.
4. S and r_R are defined as $S = \log_g \alpha$ and $r_R = \log_g \beta$ (but these values are not explicitly known).
5. Let us consider a polynomial $F(x)$ of k degree such that $F(0) = S, F(j) = s_j (j \in \Lambda_R)$. Such a polynomial $F(x)$ is uniquely determined by

$$F(x) = S\lambda_0(x) + \sum_{j \in \Lambda_R} s_j \lambda_j(x) \bmod q$$

where $\lambda_j(x)$ is a function derived by Lagrange interpolation, and defined as

$$\lambda_j(x) = \prod_{j' \in \Lambda_R,\ j' \neq j} \frac{x - j'}{j' - j} \bmod q.$$

From the above observation, it is easy to see that $g^{F(x)}$ and $g^{G_l(x)}$ can be computed using

$$g^{F(x)} = \alpha^{\lambda_0(x)} \times \prod_{j \in \Lambda_R} g^{s_j \lambda_j(x)}$$

$$g^{G_l(\psi_{lj})} = \begin{cases} g^{M_{lj}} / (g^{F(\psi_{lj})})^{r_l} & (l = 1, \ldots, R-1, j \in \Gamma_l) \\ g^{M_{lj}} / (g^{F(\psi_{lj})})^{r_R} = g^{M_{lj}} / \beta^{s_j} & (l = R,\ j \in \Lambda_R) \end{cases}$$

6. From the definition,

$$B_l \begin{pmatrix} b_{l0} \\ b_{l1} \\ b_{l2} \\ \vdots \\ b_{lk} \end{pmatrix} = \begin{pmatrix} T_l \\ G_l(\psi_{l1}) \\ G_l(\psi_{l2}) \\ \vdots \\ G_l(\psi_{lk}) \end{pmatrix} ,$$

for $l = 1, \ldots, R$, where B_l is a $(k+1) \times (k+1)$ matrix defined by

$$B_l = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & \psi_{l1} & {\psi_{l1}}^2 & \cdots & {\psi_{l1}}^k \\ 1 & \psi_{l2} & {\psi_{l2}}^2 & \cdots & {\psi_{l2}}^k \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \psi_{lk} & {\psi_{lk}}^2 & \cdots & {\psi_{lk}}^k \end{pmatrix}$$

B_l is nonsingular because it is a Vander monde matrix, so its inverse matrix ${B_l}^{-1} = \{\mu_{ij}\}_{0 \leq i,j \leq k}$ can be defined. Thus, $b_{lj} = \mu_{lj} T_l + \sum_{j'=1}^{k} \mu_{jj'} G_l(\psi_{lj'})$ $(j = 0, \ldots, k)$. From this observation, $u_{l-1,j} (j = 0, \ldots, k,\ l = 1, \ldots, R)$ is derived from

$$u_{l-1,j} = g^{b_{lj}} = g^{\mu_{lj} T_l} \times \prod_{j'=1}^{k} (g^{G_l(\psi_{lj'})})^{\mu_{jj'}}$$

for $j = 0, \ldots, k,\ l = 1, \ldots, R$. Note that $g^{G(x)}$ can be computed for $x \in \Gamma_l, l = 1, \ldots, R$ as mentioned before.

7. f_{1j} can be computed for $j \in \Lambda_R$ using

$$f_{1j} = g^{G^{(1)}(j)/F(j)} = \left(\prod_{j'=0}^{k} (g^{b_{1j'}})^{j^{j'}} \right)^{1/F(j)} = \left(\prod_{j'=0}^{k} {u_{0j'}}^{j^{j'}} \right)^{1/s_j}$$

8. $X_l = g^{r_l}$ $(l = 1, \ldots, R)$ and $u_{Rj} = g^{b_{R+1,j}}$ $(j = 0, \ldots, k)$
9. After the above setup, H_l $(l = 1, \ldots, R)$ can be represented as

$$B_l = \langle X_l \parallel \{(j, M_{lj}) | j \in \Gamma_l\} \rangle$$

$$C_l = (u_{l0} \| \ldots \| u_{lk})$$

$$H_l = (B_l \parallel C_l)$$

10. $\mathcal{M}_2$ calls $\mathcal{M}_1$ with the inputs: $(g, p, q, k, U_0, \{(s_j, f_{1j}) | j \in \Lambda_R\}, H_1, \ldots, H_R)$. If $\mathcal{M}_1$ outputs U_R, $\mathcal{M}_2$ can compute γ by $\gamma = U_R / g^{T_R}$. Thus, we can conclude $\mathcal{M}_1$ implies $\mathcal{M}_2$.

Forward Secrecy. Our protocol provides forward secrecy in a sense that an adversary $\mathcal{A}_R$ cannot compute U_R even if $\mathcal{A}_R$ can access any broadcasted information given in the future (which implies $\mathcal{A}_R$ might access future broadcast keys). Therefore, haing a key only gives information about the current round, and that there is not link between past, present and future. To prove this, we can slightly extend the above definition of M_1 to M_1^*.

Definition 2. *$\mathcal{M}_1^*$ is a probabilistic polynomial time algorithm in the size of $\langle g \rangle$, which takes $\mathcal{M}_1$'s inputs and the future headers $H_{R+1}, \ldots, H_{R'}$ for any $R' > R$, and outputs U_R with a negligible probability.*

Theorem 2. *$\mathcal{M}_1^*$ is as hard as CDH in the group $\langle g \rangle$.*

Proof. (Sketch)
The almost same proof as that to Theorem 1 can be applied. The only differnece is $\mathcal{M}_1^*$'s additional input which can be computed at step 8 and 9 by choosing r_l, M_{lj} $(j \in \Gamma_l), b_{l+1j}$ $(j = 0, \ldots, k)$ uniformly from Z_q for $l = R+1, \ldots, R'$.

6 Conclusion

This paper presented a provably secure solution to overcome a limitation of the prior broadcast exclusion protocol by resharing a blinding polynomial in a per-round pre-processing step. Our method improves the performance of updating rounds because major part of computation can be performed within the pre-processing period.

As mentioned in Section 1, we assume that the pre-processing can be completed before starting to decrypt the header for the next round. In some applications such as PPV, however, broadcast keys have to be changed frequently. Our protocol might not handle the frequent round update because it requires $O(k)$ pre-processing for each round. Reducing the pre-processing will be an issue to be solved.

Acknowledgements

The authors would like to thank Junji Shikata for his many helpful comments, and the anonymous referees for useful suggestions. We also appreciate Natsume Matsuzaki, Jun Anzai and Tsutomu Matsumoto because our result is closely built on their prior works.

References

1. Amos Fiat and Moni Naor. Broadcast encryption. In *Advances in Cryptology - CRYPTO '93*, LNCS 773, pp 480–491. Springer-Verlag, 1994.
2. Dalit Naor, Moni Naor, and Jeffrey B. Lotspiech. Revocation and tracing schemes for stateless receivers. In *Advances in Cryptology - CRYPTO 2001*, LNCS 2139, pp 41–62. Springer-Verlag, 2001.
3. Tomoyuki Asano. A revocation scheme with minimal storage at receivers. In *Advances in Cryptology - ASIACRYPT 2002*, LNCS 2501, pp 433–450. Springer-Verlag, 2002.
4. Ravi Kumar, Sridhar Rajagopalan, and Amit Sahai. Coding constructions for blacklisting problems without computational assumptions. In *Advances in Cryptology - CRYPTO '99*, LNCS 1666, pp 609–623. Springer-Verlag, 1999.
5. Jun Anzai, Natsume Matsuzaki, and Tsutomu Matsumoto. A quick group key distribution scheme with "entity revocation". In *Advances in Cryptology - ASIACRYPT '99*, LNCS 1716, pp 333–347. Springer-Verlag, 1999.
6. Yvo Desmedt and Yair Frankel. Threshold cryptosystems. In *Advances in Cryptology - CRYPTO '89*, LNCS 435, pp 307–315. Springer-Verlag, 1990.
7. Moni Naor and Benny Pinkas. Efficient trace and revoke schemes. In *Financial Cryptography 2000*, LNCS 1962, pp 1–20. Springer-Verlag, 2001.
8. Natsume Matsuzaki, Jun Anzai, and Tsutomu Matsumoto. Light weight broadcast exclusion using secret sharing. In *5th Australasian Conference Information Security and Privacy (ACISP'2000)*, LNCS 1841, pp 313–327. Springer-Verlag, 2000.
9. Kaoru Kurosawa and Yvo Desmedt. Optimum traitor tracing and asymmetric schemes. In *Advances in Cryptology - EUROCRYPT '98*, LNCS 1403, pp 145–157. Springer-Verlag, 1998.
10. Kaoru Kurosawa and Takuya Yoshida. Linear code implies public-key traitor tracing. In *5th International Workshop on Practice and Theory in Public Key Cryptosystems, PKC 2002*, LNCS 2274, pp 172–187. Springer-Verlag, 2002.
11. E. R. Berlekamp. Factoring polynomials over large finite fields. In *Math. Comp.*, pp 713–735, 1970.
12. D. E. Knuth. Seminumerical algorithms - the art of computer programming. In *Addison-Wesley.*
13. N. Courtois, A. Klimov, J. Patarin, and A. Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In *Proc. of EUROCRYPT'2000*, LNCS 1807, pp 392–407. Springer-Verlag, 2000.