Cost-effective Private Linear Key Agreement with Adaptive CCA Security from Prime Order Multilinear Maps and Tracing Traitors

Mriganka Mandal and Ratna Dutta

Department of Mathematics, Indian Institute of Technology Kharagpur, Kharagpur, 721302, India

Keywords: Private linear key agreement (PLKA) enables a group of users to agree upon a common session key in a broadcast encryption (BE) scenario, while traitor tracing (TT) system allows a tracer to identify conspiracy of a troop of colluding pirate users. This paper introduces a key encapsulation mechanism in BE that provides the functionalities of both PLKA and TT in a unified cost-effective primitive. Our PLKA based traitor tracing offers a solution to the problem of achieving full collusion resistance property and public traceability simultaneously with significant efficiency and storage compared to a sequential improvement of the PLKA based traitor tracing systems. Our PLKA builds on a prime order multilinear setting employing indistinguishability obfuscation (iO) and pseudorandom function (PRF). The resulting scheme has a fair communication, storage and computational efficiency compared to that of composite order groups. Our PLKA is adaptively chosen ciphertext attack (CCA) secure and based on the hardness of the multilinear assumption, namely, the Decisional Hybrid Diffie-Hellman Exponent (DHDHE) assumption in standard model and so far a plausible improvement in the literature. More precisely, our PLKA design significantly reduces the ciphertext size, public parameter size and user secret key size. We frame a traitor tracing algorithm with shorter running time which can be executed publicly.

1 INTRODUCTION

A private linear key agreement (PLKA) under key encapsulation framework requires the broadcaster to broadcast a common message, called header, for a specific type of user sets \[ \{i\} \in S \] where \( S = \{[1],...,[N]\} \subset 2^{|N|} \) and \([i] = \{1,...,i\}\) is the collection of users. Each user is assigned a private key by a group manager (GM). The GM is a trusted third party and the role of a broadcaster may be played by the GM or by a separate entity depending on applications. The header along with the user’s pre-assigned private key enables users in \([i]\) to extract a session key common to all the users in \([i]\). On the other hand, a PLKA based broadcast encryption (BE) empowers a content broadcaster to broadcast an encrypted message under a common session key for \([i] \in S\) so that a user \( u \in [i] \) can decrypt the ciphertext using his private key. The users outside \([i]\) obtain nothing even if they collude for both the key encapsulation model and broadcast model of PLKA. The first construction for PLKA was designed by (Boneh et al., 2006; Boneh and Waters, 2006) followed by a number of works (Garg et al., 2010; Boneh and Zhandry, 2014; Nishimaki et al., 2016).

Consider a traditional cable TV system where the broadcaster broadcasts a classified digital content encrypted under a publicly known key to a set of legitimate users. Each legitimate user, having a valid private key embedded within a set-top box provided by the GM, can successfully decrypt and recover the classified content. Any user, who has paid to get his private key from the GM, might make a reprint to resell his private key or even publish it on the Internet. This allows unauthorized users to decrypt the classified content without having a legal authorization, causing the broadcaster a massive financial loss. Consequently, the broadcaster will attempt to identify those rouge user.

A Traitor tracing (TT) system is devised to aid content broadcasters to identify conspiracy of defrauders who create a pirate decoder box. A coalition of traitors might make a conspiracy to create the pirate decoder containing an arbitrarily complex and even obfuscated malicious program and is capable of decrypting the encrypted digital content. The traitors might alter their private keys in such a way that the altered keys cannot be linked with their
original private keys. A traitor tracing system runs an efficient tracing algorithm that interacts with the pirate decoder considering it as a black-box oracle and outputs at least one identity of the traitors in the coalition who was involved to create the malicious program using his own private key. Pirate cable TV, set-top decoders, encrypted satellite radio, pirate decryption software posted on the Internet etc. are few examples of pirate decoder box.

A naive approach to address this problem is the following. For a system having \(N\) users, the broadcaster broadcasts \(N\) ciphertext under \(N\) different public keys whereby a legitimate user can decrypt the ciphertext corresponding to his own secret key. Consequently, given any pirate decoder, it is easy to pinpoint at least one traitor whose secret key is used to fabricate the pirate decoder. However, this solution is inefficient as the ciphertext size is linear in \(N\). Although a PLKA system has the capability of fraud detection, it is not always possible to switch a general BE scheme into a tracing scheme. Designing a PLKA traitor tracing, with shorter size ciphertext, public parameter and the user secret key is a challenging task.

**Related Work.** Traitor tracing was formally introduced by (Chor et al., 1994), followed by a several works in different flavors (Kiayias and Yung, 2001; Boneh and Waters, 2006; Boneh et al., 2006; Garg et al., 2010; Boneh and Zhandry, 2014; Nishimaki et al., 2016; Garg et al., 2016), and (b) secretly traceable which uses a secret tracing key to identify rogue users (Boneh et al., 2006; Kiayias and Yung, 2001). In 2014, (Boneh and Zhandry, 2014) constructed a fully collusion resistant PLKA traitor tracing with public traceability utilizing the constrained pseudorandom functions (cPRFs) and indistinguishability obfuscation (iO). All the aforementioned PLKA schemes use the Hybrid Coloring tracing approach of (Kiayias and Yung, 2001). Adopting iO, (Nishimaki et al., 2016) exhibited that a PLKA traitor tracing is an immediate consequence of functional encryption (FE). In (Garg et al., 2016), a FE scheme is designed in composite order asymmetric multilinear setting without iO and provides another indirect construction of traitor tracing. None of the schemes (Nishimaki et al., 2016; Garg et al., 2016) provide explicit construction of PLKA traitor tracing. As pointed out by (Garg et al., 2010), the communication, storage, and computational efficiency of prime order groups are much higher compared to that of composite order group. Our main focus in this work is to build a PLKA traitor tracing scheme over prime order multilinear groups (Coron et al., 2015; Gentry et al., 2015) achieving order-of-magnitude improvements in efficiency and storage without any security breach.

**Our Contribution.** We design a PLKA construction coupling pseudorandom function (PRF) of (Goldreich et al., 1986) with indistinguishability obfuscation (iO) and adopting multilinear maps over prime order group. Note that several recent attacks have broken many assumptions on known multilinear maps (Coron et al., 2015; Gentry et al., 2015). Recently, (Gu, 2015) constructed a new variant of the multilinear maps which seemed to thwart known attacks. We skillfully integrate the tracing mechanism of (Kiayias and Yung, 2001) in our PLKA, yielding the first fully collusion resistant and publicly traceable scheme.
Table 2: Comparative summary of computation and tracing time.

<table>
<thead>
<tr>
<th></th>
<th>PLKA (Boneh and Waters, 2006)</th>
<th>Pairing</th>
<th>Exponentiation</th>
<th>Product</th>
<th>Running Time of Tracing Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Boneh et al., 2006)</td>
<td>$\sqrt{N} + 4$ (bilinear)</td>
<td>$3N + (N + 15)\sqrt{N} + 4$</td>
<td>$4N + 5\sqrt{N} + 4$</td>
<td>$O(N^3)$</td>
<td></td>
</tr>
<tr>
<td>(Garg et al., 2010)</td>
<td>$\sqrt{N} + 8$ (bilinear)</td>
<td>$2N + 10\sqrt{N} + 1$</td>
<td>$N + 3\sqrt{N} + 4$</td>
<td>$O(N^2)$</td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>$2$ (multilinear)</td>
<td>$3N + 8$</td>
<td>$2N + 3$</td>
<td>$\text{poly}((\log N)^2, \eta)$</td>
<td></td>
</tr>
</tbody>
</table>

$N = \text{total number of users in the system}$, $\eta = \text{security parameter}$.

A **traceable** PLKA traitor tracing framework over prime order multilinear group setting with tracing algorithm having shorter running time. We summarize below our main findings in this work:

- Our PLKA construction significantly reduces the parameter sizes as exhibited by Table 1. The public parameter size in our construction is polylogarithmic in $N$ while the ciphertext size is logarithmic in $N$. Here, $N$ is the total number of users in the system. More interestingly, user secret key is a single multilinear group element in our PLKA.

- We emphasize that our scheme is adaptively chosen ciphertext attack (CCA)-secure under the Decisional Hybrid Diffie-Hellman Exponent (DHDHE)-assumption in standard security model and relies on iO security. Note that recently iO is aggregately constructible from the puncturable secret key functional encryption (Kitagawa et al., 2018). Our tracing algorithm enables to trace the conspiracy of an arbitrary number of defrauders using the public parameter only. On a more positive note, we have shown that although we follow the tracing approach of (Kiayias and Yung, 2001), the run time of our tracing algorithm is $\text{poly}((\log N)^2, \eta)$, where $\eta$ is the security parameter. However, the running time of tracing algorithms is $O(N^3)$ for all the existing PLKA traitor tracing schemes based on Hybrid Coloring tracing mechanism of (Kiayias and Yung, 2001). In sum, we achieve a **publicly traceable and fully collusion resistant** traitor tracing scheme with shorter running time.

- The PLKA design of (Boneh and Waters, 2006; Boneh et al., 2006; Garg et al., 2010) uses bilinear maps while that of (Boneh and Zhendry, 2014) is constructed using the security of iO and cPRFs (Boneh and Waters, 2013). The work of (Nishimaki et al., 2016; Garg et al., 2016) are based on FE. Coupling iO with the one way function, (Nishimaki et al., 2016) constructed a FE scheme and furnished an idea to transform it into a traitor tracing scheme. They set up with the exponentially large identity space and embedded user’s arbitrary information in their secret key. As a result, the user identity bit-length become arbitrarily large. As shown in Table 1, the size of ciphertext and the user secret key in their works grow with the identity bit-length which is arbitrarily large, and also the ciphertext size depends on the message-bit length. The size of the parameters in our PLKA construction are independent of identity bit-length as well as the message-bit length. Our PLKA has similar parameter sizes as that of the PLKA of (Boneh and Zhendry, 2014) which stance upon four cPRFs in generic forms showing only the input-output behavior. Additionally, the work of (Boneh and Zhendry, 2014) utilizes the multilinear map based cPRF of (Boneh and Waters, 2013) which are themselves based on multilinear maps that requires at least $O(\log N)$ symmetric multilinear pairing operations which are known to be very expensive. In contrast, we use only two PRFs of (Goldreich et al., 1986) which are efficient due to their inherent tree structures.

- Table 2 shows the computation comparison in terms of number of pairings, exponentiations, multiplications and run time of the tracing algorithm. We exclude (Garg et al., 2016; Nishimaki et al., 2016; Boneh and Zhendry, 2014) from Table 2 as suitable FE schemes and multiparty key exchange protocols are the primary requirements in these works rather than direct constructions for traitor tracing. To trace all the traitors, (Nishimaki et al., 2016) proposed an **oracle jump finding (OJF)** problem and showed that any PLKA is sufficient for traitor tracing employing OJF problem. However, to run the tracing algorithm, the works of (Nishimaki et al., 2016) requires the total number $q$ of traitors belonging to the pirate decoder $\mathcal{D}$ as an extra input and run time of OJF algorithm is $\text{poly}(\log N, q, \eta)$ which is faster than our PLKA construction. For the bounded collusion resistant schemes, $q$ is publicly known. In many real life scenarios, the tracing algorithm is given black-box interactions with $\mathcal{D}$ and finding $q$ at prior not always possible. Unlike this, our tracing algorithm does not require any prior knowledge of parameters like $q$ and runs in $\text{poly}(\log N)^2, \eta$ time using only the public parameter as the inputs.

### 2 PRELIMINARY

**Notation.** Let, $[j] = \{1, \ldots, j\}$ be the set of all positive integers from 1 to $j$. Given any set $S$, $x \in_R S$ stands for $x$ drawn uniformly at random from $S$. For a
randomized algorithm RandA, \( y \leftarrow \text{RandA}(z) \) represents output by RandA on input \( z \). The equivalence relation over a set is denoted by \( \equiv \). A probabilistic polynomial time algorithm is denoted by PPT and \( \eta \) is the security parameter.

**Definition 1. (Negligible Function)** A function \( \Psi : \mathbb{N} \rightarrow \mathbb{R} \) is said to be negligible in \( N \), if for every positive integer \( n \) there exists an integer \( N_n \) such that \( |\Psi(N)| < \frac{1}{N^n} \) for all \( N \geq N_n \).

**Definition 2. (Chernoff Bound)** Let, \( X = \sum_{i=1}^{n} X_i \), where \( X_i \) independent random variables for \( i = 1, \ldots, n \). Let \( X_i = 1 \) with probability \( p_i \), \( X_i = 0 \) with probability \( 1 - p_i \) and \( \mu = E(X) = \sum_{i=1}^{n} p_i \) is the expectation. Then, \( \Pr \{|X - \mu| \geq a\} \leq 2e^{-2a^2/\mu} \), where \( a = \mu - \delta \) is an arbitrary constant and \( 0 < \delta < 1 \).

**Definition 3. (Pseudorandom Function (PRF))** A PPT function denoted by \( \text{PRF} : \mathbb{K} \times X \rightarrow \mathbb{Y} \), that can be computed by a deterministic polynomial time algorithm which on input a fixed but randomly chosen key \( k \in \mathbb{K} \) and any point \( x \in X \), outputs \( \text{PRF}(k,x) \in \mathbb{Y} \) such that \( \text{PRF}(k,\cdot) \) is an indistinguishable function from a random function.

Henceforth, \( \text{PRF}(\cdot) \) refers to \( \text{PRF}(k,\cdot) \) for a random key \( k \in \mathbb{K} \).

**Definition 4. (Indistinguishability Obfuscator)** A uniform probabilistic polynomial time machine \( \text{iO} \) for a circuit class \( \{C_k\} \), with circuits of size \( n \) is called an indistinguishability obfuscator (iO) (Kilian et al., 2018) if it amuses the following properties.

1. Functionality Preserving: For all security parameters \( \eta \in \mathbb{N} \), for all circuit \( C \in \{C_k\} \) and for all inputs \( x, i \in \text{iO}(\eta, C) \) preserves the functionality of the circuit \( C \) under the obfuscation, i.e., \( \Pr[x, C(x) = C(x) : C \leftarrow \text{iO}(\eta, C)] = 1 \).

2. Indistinguishabilty: For all pairs of probabilistic polynomial time adversaries \( A = (D_1, D_2) \), there exists a negligible function \( \zeta(\eta) \) such that, if \( \Pr[x, C_0(x) = C_1(x) : (C_0, C_1, \sigma) \leftarrow D_1(\eta)] = 1 - \zeta(\eta) \) then \( \Pr[D_2(\sigma, i \in \text{iO}(\eta, C_0)) = 1] - \Pr[D_2(\sigma, i \in \text{iO}(\eta, C_1)) = 1] < \zeta(\eta) \).

Note that if no confusion arises, we will omit \( \eta \) as an input to \( \text{iO} \) and a subscript for \( C \).

**2.1 Asymmetric Multilinear Map and Complexity Assumption**

A (leveled) asymmetric multilinear map \( \mathcal{M} = (\mathcal{M}, \text{Setup}, \epsilon) \) of (Coron et al., 2015; Gentry et al., 2015) consists of the following two algorithms.

1. \( \mathcal{M} \) is the algorithm \( \text{Setup} \) which on input \( 1^n \) and a public key \( \rho \) outputs a \( n \)-dimensional \( \mathcal{M} \).
2. \( \epsilon \) is the description of the multilinear map, where \( \epsilon \) is a polynomial in \( \mathbb{Z} \).

**Figure 1:** \( \kappa \)-DHDHE instance generator \( g_{\mu}^{\kappa \times \text{DHDHE}} \).

**\( \kappa \)-Decisional Hybrid Diffie-Hellman Exponent Assumption (\( \kappa \)-DHDHE).** The \( \kappa \)-DHDHE problem is to guess \( \mu \in \{0,1\} \) given \( \chi_{\mu} = (e, a \text{PPM}, \Gamma_0, \ldots, \Gamma_{\kappa}, V, T_{\mu}) \) generated by the generator \( G_{\mu}^{\kappa \times \text{DHDHE}} \) shown in Figure 1.

**Definition 5. (\( \kappa \)-DHDHE Assumption)** The \( \kappa \)-DHDHE assumption is that \( \text{Adv}_{\hat{\kappa}}^{\text{DHDHE}}(\eta) \) is at most negligible for all PPT algorithms \( \mathbb{B} \).

**2.2 Hybrid Coloring**

A Hybrid Coloring of the user population, introduced by (Kiayias and Yung, 2001), is a partition of the total population into two groups, where one group is color-coded as \( \mathbb{A} \) and the other as \( \mathbb{B} \). The coloring is such that there exists a PPT mapping from \( \mathbb{A} \) to \( \mathbb{B} \) or vice versa. This mapping ensures that the coloring is fair and secure against cheating.
number of users $|N|$ in a broadcast encryption (BE) system. A random ciphertext $C_R$ induces a Hybrid Coloring over $|N|$ as follows.

- Let $D$ be a pirate decoder (PD) box. We define an equivalence relation over the user secret key space as follows: $\forall u, u' \in |N|$, $p_k_u \equiv p_k_{u'}$ iff $Pr[D(1^1, p_k_u, C_R) \neq D(1^1, p_k_{u'}, C_R)] \leq \epsilon$, where $\epsilon$ is a negligible quantity and $p_k_u$ and $p_k_{u'}$ are the secret key of $u$ and $u'$ respectively.
- Assume that $C_m$ be a ciphertext corresponding to a valid message $m$. Then, with overwhelming high probability $D(1^1, p_k_u, C_m) = D(1^1, p_k_u, C_m)$ for all $u, u' \in |N|$. In that case, we get a unique equivalence class. Consequently, all the users will get the same color. Let $\text{Ciph}_{pr}$ be the set of all random ciphertexts such that for all $C' \in \text{Ciph}_{pr}, C'$ induces a unique equivalence class. Then, the set of all valid ciphertexts constitute a subset of $\text{Ciph}_{pr}$.
- A BE scheme induces a Hybrid Coloring if there exist an algorithm that produces a ciphertext $C$ such that $C$ induces a partition over the user population.

One important observation regarding the tracing algorithm of (Kiayias and Yung, 2001) is formally stated by the following lemma.

**Lemma 1.** (Kiayias and Yung, 2001) The tracing procedure using the Hybrid Coloring has time complexity $O(N^2 \log^2 N)$ and identify a traitor with high probability.

## 3 OUR PLKA TRACING SCHEME

Our PLKA consists of three randomized algorithms PLKA.Setup, PLKA.Enc, PLKA.Dec and an external tracing algorithm PLKA.Trace $^D$ which are described below.

- $(\text{plparams}, (\text{plsk}_1, \ldots, \text{plsk}_N)) \leftarrow \text{PLKA.Setup}(\eta, \kappa)$: The group manager (GM) takes as input the length $\kappa$ of the identities along with the security parameter $\eta$ and proceeds as follows. The identity space is $ID = \{0, 1\}^\kappa \setminus \{0^\kappa\}$ and the total number of users the system can support is $N = (2^\kappa - 1)$.
  (i) The GM first constructs $\tilde{\rho} = (1, \ldots, 1)$, a $(\kappa + 1)$-length vector with all 1’s, and runs the setup algorithm $a\text{M.Setup}(1^1, 2\tilde{\rho})$ for the multilinear map described in section 2.1 to generate the public parameter $a\text{PPM} = (\kappa, g_{\xi}, \ldots, g_{\eta})$ where $g_{\xi}$ is the canonical generator of the $i$-th source group $G_{\xi}$ for $0 \leq i \leq \kappa$ and $G_{\eta}$ is the target group. All the groups have the same large prime order $p > 2^\kappa$. It generates the canonical generators $g_\beta$ and $C_{\beta}$ of the groups $G_\beta$ and $G_{\eta}$ respectively by the repeated multilinear pairing operations using $a\text{PPM}$.

(ii) Two GGM tree (Goldreich et al., 1986) based secure pseudorandom functions $\text{PRF}_{\text{rand}} : \{0, 1\}^{2n} \rightarrow \{0, \ldots, N\}$ and $\text{PRF}_{\text{auth}} : \{0, 1\}^{2n} \times |N| \rightarrow \{0, 1\}^n$ are selected by the GM where rand, auth are keys randomly chosen from the key space $\mathcal{K} = \{0, 1\}^n$. It also picks $\text{PRG} : \{0, 1\}^n \rightarrow \{0, 1\}^{2n}$, the length doubling pseudorandom generator (Blum and Micali, 1984).

(iii) The GM chooses $\xi, \tau \in R Z_p$, sets the programs $\text{PT}_{\text{Enc}}$ (Figure 2), $\text{PT}_{\text{Dec}}$ (Figure 3) and obfuscate these to generate obfuscated programs $\text{PT}_{\text{Enc}} = iO(\text{PT}_{\text{Enc}}), \text{PT}_{\text{Dec}} = iO(\text{PT}_{\text{Dec}})$ respectively using a secure indistinguishability obfuscator $iO$. The program $\text{PT}_{\text{Enc}}(j \in |N|, t \in \mathbb{Z}_p, s \in \{0, 1\}^n)$ has $(\text{PRF}_{\text{rand}}, \text{PRF}_{\text{auth}}, (\xi, \tau), \kappa, g_\beta, g_{2\beta})$ hard-coded in it and runs on input $j, t, s$ to generate a header-session key pair $\langle \text{Hdr} = (r \in \{0, 1\}^n, C_1 \in \{0, 1\}^n, C_2 \in \{0, 1\}^n, C_3 \in G_\beta, C_4 \in G_{\eta} \rangle, K_{\text{PLKA}} = (g_{2\beta})^{\kappa r} \rangle$.

- $(\eta, \kappa)$: The program $\text{PT}_{\text{Dec}}$.

![Figure 2: The program $\text{PT}_{\text{Enc}}$.](image-url)

On the other hand, the program $\text{PT}_{\text{Dec}}(\text{Hdr}, u \in |N|, \text{plsk}_u \in G_{\eta})$ has $\text{PRF}_{\text{rand}}, \text{PRF}_{\text{auth}}, (\xi, \tau), \kappa, g_\beta, g_{2\beta}$ hard-coded in it and runs on inputs $\text{Hdr}, u, \text{plsk}_u$ to generate the correct session key $K_{\text{PLKA}}$. The obfuscated programs $\text{PT}_{\text{Enc}}$ and $\text{PT}_{\text{Dec}}$ behave in a similar manner as $\text{PT}_{\text{Enc}}$ and $\text{PT}_{\text{Dec}}$ respectively. That is, on the same input, $\text{PT}_{\text{Enc}}$ and $\text{PT}_{\text{Enc}}$ generate the same output. Similarly, $\text{PT}_{\text{Dec}}$ and $\text{PT}_{\text{Dec}}$ provide the same output on the same input. Note that in step 1(b) of $\text{PT}_{\text{Enc}}$, from the GGM tree based construction $\text{PRF}_{\text{rand}}(r)$ is an $\eta$-bit string which is converted to an integer and added to $j$ modulo $(N + 1)$ to generate header component $C_2$. Similarly, in step 1(a) of $\text{PT}_{\text{Dec}}$ to recover $j$ from the header component $C_2$ we consider the integer representation of the $\eta$-bit string $\text{PRF}_{\text{rand}}(r)$.

(iv) The GM finally publishes the private linear public parameter $\text{plparams} = (\text{PRF}_{\text{rand}}, \text{PRF}_{\text{auth}},$...
Inputs: $\text{Hdr} = (r \in \{0,1\}^{2\eta}, \ C_1 \in \mathbb{N}, \ C_2 \in \{0,1\}^{3\eta}, \ C_3 \in \mathbb{G}_p, \ C_4 \in \mathbb{G}_p), \ u \in \mathbb{N}, \ plsk_u \in \mathbb{G}_p$

Constants: $\text{PRF}_{\text{rand}}, \text{PRF}_{\text{auth}}, (\xi, \tau), \ k, \ g_p, g_{2p}$

1. Compute:
   (a) $j = (C_1 - \text{PRF}_{\text{rand}}(r)) \mod (N + 1)$
   (b) $x = \text{PRG}(\text{PRF}_{\text{auth}}(r, C_1))$
   (c) $y = (g_p)^{2u}$

2. Check that $(u \leq j) \land (x = \text{PRG}(C_2)) \land (y = \text{plsk}_u)$
   (a) If check fails, output $\bot$ and stop
   (b) Otherwise, compute:
      i. $\Lambda_2^{x-i+u} = (g_p)^{2x-i+u}$ for all $i \in [j], i \neq u$
      ii. $\Lambda_u = (g_p)^{2u}$
   (c) $\text{KL}_{\text{PLKA}} = \frac{e(\Lambda_u, C_4)}{e(plsk_u, \prod_{i=x-i+u}^j \Lambda_2^{x-i+u}, C_3)}$

3. Output: $\text{KL}_{\text{PLKA}}$

Figure 3: The program $\text{PT}_{\text{Dec}}$

PRG, $\text{PT}_{\text{Enc}}, \text{PT}_{\text{Dec}}$. For each user $u \in \mathbb{N}$, it computes the user secret key $\text{plsk}_u = (g_p)^{2u}$ and sends $\text{plsk}_u$ to user $u$ through a secure communication channel between the GM and the user $u$.

1. $(\text{Hdr}, \text{KL}_{\text{PLKA}}) \leftarrow \text{PLKA.Enc}(\text{plparams}, j \in \mathbb{N})$: On input an integer $j \in \mathbb{N}$ and the public parameter $\text{plparams}$, the encryptor executes the following steps:
   (i) It chooses elements $t \in \mathbb{Z}_p$ and $s \in \{0,1\}^n$.
   (ii) It generates $(\text{Hdr} = (r, C_1, C_2, C_3, C_4), \text{KL}_{\text{PLKA}})$ by running the program $\text{PT}_{\text{Enc}}$ extracted from $\text{plparams}$, on input $(j \in \mathbb{N}, t \in \mathbb{Z}_p, s \in \{0,1\}^n)$, where $\text{Hdr} = (r, C_1, C_2, C_3, C_4)$ is the ciphertext header and $\text{KL}_{\text{PLKA}}$ is the session key for all the users in the set $[j]$.
   (iii) Finally, it publishes $\text{Hdr}$ as the ciphertext and keeps $\text{KL}_{\text{PLKA}}$ as secret to itself.

2. $(\text{KL}_{\text{PLKA}}, \bot) \leftarrow \text{PLKA.Dec}(\text{plparams}, u \in \mathbb{N}, \text{plsk}_u, \text{Hdr} = (r, C_1, C_2, C_3, C_4))$: A user $u \in \mathbb{N}$ uses secret key $\text{plsk}_u = (g_p)^{2u}$ to recover the session key $\text{KL}_{\text{PLKA}}$ from the ciphertext header $\text{Hdr} = (r, C_1, C_2, C_3, C_4)$ as follows:
   (i) It runs the program $\text{PT}_{\text{Dec}}$, extracted from $\text{plparams}$, on input $(\text{Hdr} = (r, C_1, C_2, C_3, C_4), u, \text{plsk}_u)$.
   (ii) If it passes all the checking conditions in step 2 of the program $\text{PT}_{\text{Dec}} = iO(\text{PT}_{\text{Dec}})$ in Figure 3, it proceeds as follows.

Algorithm 1: Traitor tracing program $\text{Trace}^{\theta}$

1. Input: $\text{plparams}, \varepsilon$
2. For $i = 0$ to $N$
3. success $\leftarrow 0$
4. For $j = 1$ to $2 \left(\frac{\log N}{\varepsilon}\right)^2$
5. $(\text{Hdr}^{(j)}, \text{KL}_{\text{PLKA}}^{(j)}) \leftarrow \text{PLKA.Enc}(\text{plparams}, i)$
6. $\text{KL}_{\text{PLKA}}^{(j)} \leftarrow \text{D}(\text{Hdr}^{(j)})$
7. If $\text{KL}_{\text{PLKA}}^{(j)} = \text{KL}_{\text{PLKA}}^{(i)}$ then
8. success $\leftarrow$ success + 1
9. End if
10. End for
11. $\gamma_{\text{obsrv}} \leftarrow \text{success}$
12. End for
13. Return $T^{\text{TTS}} = \{i : \gamma_{i}^{\text{obsrv}} - \gamma_{i-1}^{\text{obsrv}} \geq \frac{4(\log N)^2}{\varepsilon}\}$

gets the correct key $\text{KL}_{\text{PLKA}}$ as the output; otherwise gets $\bot$.

- $T^{\text{TTS}} \leftarrow \text{PLKA.Trace}^{\theta}(\text{plparams}, \varepsilon)$: The tracer takes as input the public parameter $\text{plparams}$, a parameter $\varepsilon$ which is polynomially related to the security parameter $\eta$. It runs the $\text{Trace}^{\theta}$ program of Algorithm 1, on input the public parameter $\text{plparams}$ and the parameter $\varepsilon$. It outputs the set of users $T^{\text{TTS}} \subseteq \{1, \ldots, N\}$ as the traitor users.

Correctness and the proof of our tracing algorithm is shown in the Theorem 2.

Correctness. Let $u$, $j \in \mathbb{N}$ and $1 \leq u \leq j \leq N$. Let, $(\text{plparams}, (\text{plsk}_1, \ldots, \text{plsk}_u))) \leftarrow \text{PLKA.Setup}(\eta, \kappa)$, where $\text{plparams}= (\text{PRF}_{\text{rand}}, \text{PRF}_{\text{auth}}, \text{PRG}, \text{PT}_{\text{Enc}}, \text{PT}_{\text{Dec}})$ and $\text{plsk}_u = (g_p)^{2u}$. Let $(\text{Hdr}, \text{KL}_{\text{PLKA}} = (g_p)^{2x}) \leftarrow \text{PLKA.Enc}(\text{plparams}, j \in \mathbb{N})$, where $\text{Hdr} = (r, C_1, C_2, C_3, C_4)$ with

$$C_1 = (\text{PRF}_{\text{rand}}(r) + j) \mod (N + 1), \ C_3 = (g_p)^{2j},$$
$$C_2 = \text{PRF}_{\text{auth}}(r, C_1), \ C_4 = (g_p)^{\sum_{i=1}^{j+1} 2^{2x-i}}.$$

A user $u$, with its secret key $\text{plsk}_u = (g_p)^{2u}$ runs $\text{PLKA.Dec}(\text{plparams}, u, \text{plsk}_u, \text{Hdr})$. If $u$ passes all the conditions in step 2 of the program in Figure 3 in executing the program $\text{PT}_{\text{Dec}}$ in $\text{plparams}$, then we show below that $u$ can recover the correct session key $\text{KL}_{\text{PLKA}} = (g_p)^{2x}$ by extracting $C_3$ and $C_4$ from $\text{Hdr}$ and proceeding as follows.

As, $\Lambda_2^{x-i+u} = (g_p)^{2x-i+u}$ and $\Lambda_u = (g_p)^{2u}$ are gi-
ven in $PT_{\text{Dec}}$, we have
\[
e(\Lambda_u, C_4) \left/ \mathcal{e}\left(\sum_{j=1}^{\ell} (g_\rho_j)^{x_j}, (g_\rho_j)^{\sum_{i=1}^{\ell} x_{i-1}}\right)\right/ e\left(\sum_{j=1}^{\ell} (g_\rho_j)^{x_j}, (g_\rho_j)^{\sum_{i=1}^{\ell} x_{i-1}}\right)
\]
\[
= e\left(\prod_{i \neq u} \Lambda_{2^k-i+2}, C_3\right)
\]

**Remark 1.** As the set system $\mathcal{S} = \{\mathcal{I}, \ldots, \mathcal{N}\}$ has only a polynomial number of recipient sets in it, according to (Boneh and Zhandry, 2014), the selective and the adaptive security are equivalent.

## 4 SECURITY ANALYSIS

**Theorem 1. (Security of Indistinguishability)** Assuming secure iO, our PLKA scheme, presented in section 3, achieves adaptive CCA-security under the $\kappa$-DH/DHE assumption.

**Proof.** Due to limited space, proof is available in the full version (Mandal and Dutta, 2018).

**Theorem 2. (Security of Traceability)** Suppose that our PLKA scheme, presented in section 3, is adaptive CCA-secure. Then, the publicly traceable PLKA.Trace$^D$ algorithm outputs identity of all the traitors.

**Proof.** Assume that at the beginning the adversary $\mathcal{A}$ outputs a pirate decoder box $\mathcal{D}$. For $i = 0, \ldots, \mathcal{N}$ construct the experiment TrExp. Of Figure 4 using the Hybrid Coloring mechanism shown in section 2.2. Let $p_i = \Pr[H_{\mathcal{H}_i} = \text{success}]$ be the success probability in the above experiment TrExp, for $i = 0, \ldots, N$. Clearly, $p_0 = 0$, whereas $p_N = 1$ and hence $|p_N - p_0| = 1$.

Consider that user $j \in \mathcal{N}$ is not a traitor user. Then, the secret key plsk$_j$ of user $j$ is not embedded into the pirate decoder box $\mathcal{D}$. Note that if plsk$_j$ is embedded into $\mathcal{D}$ for some $k < j$, then $H_{\mathcal{H}_j} = \mathcal{H}_k = \text{success}$ and consequently $|p_j - p_k| = 0$. On the other hand, if $j \in \mathcal{N}$ is the least positive integer such that plsk$_j$ is embedded into $\mathcal{D}$, then $H_{\mathcal{H}_j} = \text{success}$ but $H_{\mathcal{H}_k} = \text{failure}$ for $1 \leq k \leq j - 1$. In this case, $|p_j - p_k| \geq \frac{1}{N}$. However, the adversary $\mathcal{A}$, who has formed the pirate decoder box $\mathcal{D}$, can not distinguish the ciphertext headers $\mathcal{H}(i)$ and $\mathcal{H}(j-1)$ without having the knowledge of plsk$_j$, even if $\mathcal{A}$ has the secret key plsk$_j$ for $1 \leq k \leq j - 1$. As a result, the difference between the success probability in the experiment TrExp$_j$ and in the experiment TrExp$_{i-1}$ is negligible in the total number of user $N$. Therefore, $|p_j - p_{j-1}|$ is negligible in $N$.

(i) The tracer generates header-session key pair $(\mathcal{H}(i), \mathcal{K}_{\text{PLKA}}(i)) \leftarrow \text{PLKA.Enc}(\text{plparams}, i)$, where plparams is the public parameter generated using PLKA.Setup algorithm of our PLKA scheme.

(ii) Then, tracer interacts with the pirate decoder $\mathcal{D}$, giving $\mathcal{H}(i)$ as an input to $\mathcal{D}$, and in return tracer will get $\mathcal{K}_{\text{PLKA}}(i) \leftarrow \mathcal{D}(\mathcal{H}(i))$. Here, $\mathcal{D}$ acts as a black-box oracle for this interaction.

(iii) Finally, tracer sets the success or failure $\mathcal{H}_t$ as follows

\[
\mathcal{H}_t = \begin{cases} 
\text{success} & \text{if } \mathcal{K}_{\text{PLKA}}(i) = \mathcal{K}_{\text{PLKA}}(i) \\
\text{failure} & \text{otherwise}
\end{cases}
\]

Figure 4: Tracing Experiment TrExp, for $i = 0, \ldots, N$.

Since $|p_N - p_0| = 1$, by the triangular inequality there must exists at least one user $i_j \in \mathcal{N}$ such that $|p_j - p_{i_j-1}| \geq \frac{1}{N}$. So that the success probability difference between the two experiments TrExp$_i$ and TrExp$_{i-1}$ is at least $\frac{1}{N}$ which is non-negligible. Let the advantage of breaking the indistinguishability security of our PLKA scheme is $\varepsilon = \text{Adv}_{\mathcal{D}}^{\text{CA-PLKA}}(\eta)$. If $|p_0 - p_{i_j-1}| \geq \frac{1}{N}$, then this indicate that plsk$_i$ is embedded into $\mathcal{D}$ with probability at least $\varepsilon$ and hence the user $i_j$ must be a traitor. Observe that user $i_j - 1$ can not be a traitor. If both $i_j$ and $i_j - 1$ are traitors, then $\mathcal{H}_0 = \text{success}$ as well as $\mathcal{H}_{i_j - 1} = \text{success}$, as $\mathcal{D}$ having plsk$_{i_j-1}$ can return correct session keys corresponding to both $\mathcal{H}(i_j)$ and $\mathcal{H}(i_j-1)$. Note that $\mathcal{D}$ can decrypt the ciphertext header $\mathcal{H}(i_j)$ for any $j > i_j - 1$ if plsk$_{i_j-1}$ is embedded in $\mathcal{D}$.

To ensure perfectly that the user $i_j$ is a traitor user, one has to repeat the experiment TrExp$_i$ more than a single time. Consider that for each $i = 0, \ldots, N$, the tracer repeats the experiment TrExp$_i$ independently up to $\mathcal{R}$ trials. We define a random variable $\mathcal{R}'$ as total number of success that were returned by $\mathcal{D}$ during $\mathcal{R}$ trials of the experiment TrExp$_i$. If $i_j$ is a traitor user, then for one trial $|p_{i_j} - p_{i_j-1}| \geq \varepsilon$. Therefore, for $\mathcal{R}$ trials the expected difference between the random va-
riable $\mathcal{Y}_t$ and $\mathcal{Y}_{t-1}$ is at least $\varepsilon \mathbb{R}$. To perfectly ensure that the user is a traitor user, we have to make sure that the observed values of the random variables $\mathcal{Y}_k$, denoted by $\mathcal{Y}_k^{\text{obsrv}}$, is sufficiently closed to their expected values $\mu_k = p_k \mathbb{R}$ for $k = t, t - 1$. Using the Chernoff bound, we obtain the following relation between $\mathcal{Y}_k^{\text{obsrv}}$ and its expected value $\mu_k = p_k \mathbb{R}$ for $k = t, t - 1$, taking $\delta = \frac{1}{4}$, and setting $a = \frac{\varepsilon \mathbb{R}}{2}$:
\[
Pr \left( \left| \mathcal{Y}_k^{\text{obsrv}} - \mu_k \right| \geq \frac{\varepsilon \mathbb{R}}{2} \right) \leq 2 \left( e^{\frac{-\varepsilon^2}{2a}} \right) = 2 \left( e^{\frac{-\varepsilon^2}{2 \frac{\varepsilon \mathbb{R}}{2}}} \right) \leq 2N^{-\log N}
\]
if $\mathbb{R} \geq 2 \left( \frac{\log N}{k} \right)^2$. Observe that this probability is negligible in $N$ using the Definition 1, as $\log N$ is an positive function.

Again from the Chernoff bound, we can write $\mu_k = \frac{2 \varepsilon \mathbb{R}}{2} \geq \mu_k^{\text{obsrv}} \geq \mu_k + \frac{2 \varepsilon \mathbb{R}}{4}$ and $-\mu_k^{\text{obsrv}} \geq -\mu_k + \frac{2 \varepsilon \mathbb{R}}{4}$. If $i$ is a traitor, then for $i_1$ and $i_1 - 1$, the difference between two observed variables $\mathcal{Y}_1^{\text{obsrv}}$ and $\mathcal{Y}_{i_1-1}^{\text{obsrv}}$ (repeat each up to $\mathbb{R}$ times) is given by
\[
(\mathcal{Y}_1^{\text{obsrv}} - \mathcal{Y}_{i_1-1}^{\text{obsrv}}) \geq (\mu_1 - \mu_{i_1-1}) \geq \mathbb{R} + (p_{i_1} - p_{i_1-1}) \mathbb{R} \geq 2 \varepsilon \mathbb{R}
\]
Hence, for the traitor user $i_1$, the difference between $\mathcal{Y}_1^{\text{obsrv}}$ and $\mathcal{Y}_{i_1-1}^{\text{obsrv}}$ is at least $2 \varepsilon \mathbb{R}$, where $\mathbb{R} \geq 2 \left( \frac{\log N}{k} \right)^2$. The complete tracing mechanism is given in Algorithm 1.

5 CONCLUSION

In this work, we have designed an adaptively CCA-secure PLKA traitor tracing scheme, under the prime order multilinear group setting, which is fully collusion resistance and publicly traceable. Our construction is proven to be secure under the hardness of standard DHDHE-assumption. More precisely, our design significantly reduces the parameter sizes and the tracing time which are so far a plausible improvement in the literature.

REFERENCES