Generic Trace and Revoke Scheme*

Murat Ak[1], **Aggelos Kiayias**[2], **Serdar Pehlivanoglu**[3], **Ali Aydin Selcuk**[4]

[1] *Bilkent University, Turkey* muratak@cs.bilkent.edu.tr
[2] *University of Connecticut, CT, USA* aggelos@cse.uconn.edu
[3] *Zirve University, Turkey* serdar.pehlivanoglu@zirve.edu.tr
[4] *Bilkent University, Turkey* selcuk@cs.bilkent.edu.tr

Extended Abstract

1 Introduction.

Broadcast encryption (BE) is a cryptographic primitive that allows a broadcaster to encrypt a content to a specific group of users called *privileged* users and prevent *revoked* users from decrypting the content even if they collude [6]. In BE schemes, a group of users, called *traitor*s may collude and form a *pirate decoder*. In order to trace such actions, *Traitor tracing* (TT) schemes are proposed so that users that contribute to such decoders can be detected and their keys get revoked [4]. A challenging problem is to design *trace and revoke* (T&R) systems which combines the functionalities of both BE and TT schemes.

In this paper, we propose a generic method to obtain T&R schemes from BE schemes. Our construction preserves the main efficiency parameters of the underlying BE schemes. We illustrated this fact by instantiating our construction with some known BE schemes. These sample instances outperform the existing T&R schemes in either the private key size or the ciphertext length. Of a particular interest, we note that one of these instantiations yields the first identity based T&R scheme.

2 Background and Definitions

Broadcast Encryption: BE schemes are mostly used to encrypt large digital media. Therefore encrypting the whole content with the BE scheme directly is not a good idea. Typically, the data is encrypted with a symmetric encryption scheme while the symmetric key used is encrypted with the BE scheme, which is called key and data encapsulation mechanisms (KEM and DEM respectively). So, a BE scheme, defined in a KEM-DEM structure, consists of the algorithms below: Let $[n]$ denote the set of all users, $\{1, 2, \ldots, n\}$.

- `KeyDist(`1^n`)` outputs user private keys sk_i, $i \in [n]$ and the public key, PK.

*This work is supported in part by the Turkish Scientific and Technological Research Agency (TUBITAK), under grant number 111E213.

- `Encrypt(`PK, S`)` given $S \subseteq [n]$, outputs a header hdr and a symmetric key K using PK. K is the key to be used in the DEM phase. hdr is broadcast so that privileged users can use it to recover K.

- `Decrypt(`PK, i, sk_i, S, hdr`)` is run by the receiver i. If $i \in S$, that is the set used to produce hdr, then it must return K. Otherwise, it returns $\perp$.

A BE scheme is *correct* if $\forall PK, \forall S \subseteq [n], \forall i \in S$, `Decrypt(`$PK, sk_i, S, hdr$`)`$= K$, whenever $(PK, sk_1, \ldots, sk_n) \leftarrow$ `KeyDist(`1^n`)` and $(hdr, K) \leftarrow$`Encrypt(`PK, S`)`. Confidentiality is satisfied if no collusion of revoked users gets any information about the message with non-negligible probability.

Trace and Revoke Schemes: A trace and revoke (T&R) system is a system which has both tracing and revoking capabilities: *tracing* is defined as detecting users who leak their decryption keys, and *revoking* is invalidating the keys of such users. It shares the basic three algorithms of the broadcast encryption defined above to support revocation. However, for the clarity of the distinction of the notions, we will rename the algorithms as (`Setup(`1^n`)`, `Transmit(`PK, S`)`, `Receive(`PK, i, sk_i, S, hdr`)`). We again follow the definition for KEM design.

We will consider black box tracing, where we assume that we interact with a pirate decoder in a black box manner, i.e., we can only make decryption queries to the pirate decoder without seeing its inside. So, three necessary assumptions on the pirate decoder are (i) resettability: it does not maintain state, (ii) availability: it remains available as long as the tracing process continues, and (iii) usefulness: it succeeds in decrypting ciphertexts intended for at least one subset with a non-negligible probability. Therefore, we call a decryption box $\mathcal{D}_S^\sigma$ a (σ, S)-pirate if its rate of correctly decrypting broadcasts to set S is at least σ. The formal definition of the Trace algorithm of a T&R system is given below.

- `Trace(`$S, \mathcal{D}_S^\sigma, PK, TK$`)` takes a set S and a pirate $\mathcal{D}_S^\sigma$, using PK and TK (which is the tracing secret key produced in `Setup()`), it outputs a set A of accused traitors whose key(s) must have contributed to $\mathcal{D}_S^\sigma$.

Confidentiality of a T&R system is the same as that of BE schemes. A T&R system satisfies traceability if the probability that an attacker cannot be traced is negligible.

Fingerprinting Codes: The fingerprinting codes are the basic mathematical tools employed in tracing systems. It will also be main tool in our construction. We refer to the first chapter of [10] for an introductory discussion on fingerprinting codes and we give a very brief definition for fingerprinting codes:

A codeword x over an alphabet Q (with $|Q| = q$) is an ℓ-tuple $x_1, \ldots, x_\ell$ where $x_i \in Q$ for $1 \leq i \leq \ell$. We call a set of codewords $\mathcal{C} \subseteq Q^\ell$ with size n, an (ℓ, n, q)-code. Given an (ℓ, n, q)-code $\mathcal{C}$, each codeword $x \in \mathcal{C}$ is considered as a unique fingerprint of a receiver that has access to an object/functionality through its codeword.

Fingerprinting codes are defined by two algorithms, (`CodeGen`, `Identify`). The first algorithm `CodeGen(`1^n`)` outputs a pair $(\mathcal{C}, tk)$ where $\mathcal{C}$ is an (ℓ, n, q)-code with alphabet Q such that $|Q| = q$, and tk is a key for identifying purposes (possibly empty). `Identify(`$\mathcal{C}, tk, c$`)` is an algorithm which takes a code $\mathcal{C}$, a string $c \in Q^\ell$, and identifying key tk and outputs a codeword index t or $\perp$. The performance of fingerprinting codes is

evaluated according to their capability of identifying traitor codewords. Namely, a fingerprinting code that is capable of identifying a traitor codeword constructed by a coalition of size w with failure probability at most α is called an (α, w)-identifier fingerprinting code. When the failure probability $\alpha = 0$, we say it is a w-*identifier* fingerprinting code. If we also have $w = n$, we call it a fully collusion resistant fingerprinting code. In this work, we will employ binary fingerprinting codes.

3 Generic T&R scheme

In this section we discuss our contribution: efficient and effective design of a generic trace and revoke scheme from any given BE scheme `B` by making use of its algorithms (`BKeyDist`, `BEncrypt`, and `BDecrypt`).

The Main Idea: Whenever a BE transmission is to be made to a set A, instead of encrypting directly for A, we first partition A into two subsets B and C and encrypt for both separately.

As part of the black-box tracing, we will query the pirate decoder with some special ciphertexts (that we call tracing ciphertexts) and will infer some partial information on the traitor keys available to the pirate decoder based on the responses we get from the pirate decoder. The tracing ciphertexts will simply look like the regular transmissions with the exception that a random message is encrypted to C instead of the real message. If the pirate box turns out to be successful in decrypting such tracing ciphertext, we infer that a pirate key used to forge that pirate box belongs to a traitor located in set B. If it does not decrypt to the correct message then we infer that a pirate key used to forge that pirate box belongs to a traitor located in set C. Over a sequence of tracing transmissions for different choices of B and C, the tracer collects information on the traitor locations. The tracing partition will be based on a binary fingerprinting code so that it is possible to locate a traitor identity after tracing ciphertexts as many as the length of the code.

In this direction, the center sets up a fingerprinting code `F` = (`CodeGen`, `Identify`) and generates an $(\ell, n, 2)$-fingerprinting code $\mathcal{C} = \{c_1, \ldots, c_n\}$ in the key distribution phase: the code defines the sets $S_j = \{i : c_i[j] = 1\}$, for bit positions $j = 1, \ldots, \ell$. As part of the tracing process, the pirate decoder will be given a pair of BE encryptions, for each $j \in [\ell]$ intended for sets $S \cap S_j$ and $S \setminus S_j$ where the latter set of receivers are actually given a random message. If the pirate box decrypts to the actual message, we infer that the pirate key used to forge that pirate box has a 1 (resp. 0) in that position, i.e. a traitor is contained in set S_j (resp. $[n] \setminus S_j$). We construct a pirate codeword by marking 1 (resp. 0) in the j-th position of a codeword of length ℓ. When we go through all bit positions, we end up with a pirate codeword of length ℓ. That codeword will enable us to identify a traitor by calling the identifying algorithm of the fingerprinting code. We also take extra care to generate the tracing ciphertexts to be indistinguishable from regular ciphertexts.

Formal description of the generic construction:

- `TRKeyDist(`1^n`)` The key distribution algorithm `TRKeyDist(`1^n`)` of `T` runs the key distribution algorithm `BKeyDist(`1^n`)` of `B`.

  This will produce a public key PK and a set of private keys sk_i, $i \in [n]$, which will be distributed to the receivers. It chooses description of a binary fingerprinting code

F = ($\mathtt{CodeGen}()$, $\mathtt{Identify}()$) that is publicly samplable by $\mathtt{Z}()$. Here, we note that the actual codewords are not generated at this moment but the code length ℓ is pre-computed based on n and the security parameters. Hence we do not require any tracing key while the algorithm $\mathtt{Z}()$ and the length ℓ are part of the public key.

- $\mathtt{Transmit}((PK, \mathtt{Z}(), \ell), S)$ The algorithm $\mathtt{Z}(1^n, j)$ first samples a subset $U \subseteq [n]$ for a randomly chosen $j \in [\ell]$. It sets $S_1 = U \cap S$ and $S_2 = S \setminus (S \cap U)$ that partitions S into two (i.e., $S = S_1 \cup S_2$ and $S_1 \cap S_2 = \emptyset$). A key k is chosen to be transmitted by the scheme(recall we consider the KEM design, hence we transmit the key to be used in the DEM design). The transmission algorithm then runs the encryption algorithm of the BE scheme for both subsets and broadcasts the message $c = (c_1||c_2)$ which is obtained as

$$(hdr_1, K_1) \leftarrow \mathtt{BEncrypt}(PK, S_1), \quad (hdr_2, K_2) \leftarrow \mathtt{BEncrypt}(PK, S_2)$$
$$c_1 \leftarrow hdr_1||SymEnc_{K_1}(k), \quad c_2 \leftarrow hdr_2||SymEnc_{K_2}(k)$$

- $\mathtt{Receive}(PK, sk_i, S, c)$ When user i receives a ciphertext $c = (c_1||c_2)$, it first parses hdr_1 and hdr_2 from c_1 and c_2, respectively. Then it parses S_1 and S_2 from these headers. (We note that depending on the underlying broadcast encryption scheme, trial-and-error in decryption may suffice instead of specifying S_1 or S_2 explicitly) Then it uses the decryption function of B to decrypt both ciphertext parts as $\mathtt{BDecrypt}(PK, sk_i, S_1, hdr_1)$ and $\mathtt{BDecrypt}(PK, sk_i, S_2, hdr_2)$. Since a user $u \in S$ is either in S_1 or S_2, it will be able to extract either K_1 or K_2 which will enable the recovery of the key k.

- $\mathtt{Trace}(\mathcal{D}_S^1, (PK, \mathtt{Z}(), \ell), TK)$ As part of the tracing process we will query the pirate decoder $\mathcal{D}_S^1$ (we consider only the case for $\sigma = 1$ here and put remarks on $\sigma < 1$ cases later in the section) with specially designed tracing ciphertexts and observe the responses of the pirate decoder. The tracing ciphertexts are different from regular ciphertexts in two ways: (i) Instead of using the partition based on the sampler $\mathtt{Z}(1^n, j)$ for randomly chosen $j \in \ell$, we generate a binary code of length ℓ by running $\mathcal{F}.\mathtt{CodeGen}(1^n)$ and use the partition based on the j-th column(we do this for every $j \in [\ell]$) of the generated code (ii) Rather than encrypting the message for both subsets in the partition, we encrypt a random message to one of them. The complete description of the tracing procedure is summarized in the following algorithm:

Having the following theorem, we will provide the detailed proof and parameter analysis in the full version of the paper:

Theorem. *Let B be a BE scheme that is KEM-IND-CCA secure, and SYM be a symmetric encryption scheme that is IND-CCA secure then the trace and revoke scheme designed above also satisfies KEM-IND-CCA security.*

Let F, be an (ϵ_f, t)-identifier fingerprinting code, which can be publicly samplable by sampler $\mathtt{Z}()$, then for any subset $S \in [n]$, a pirate decoder $\mathcal{D}_S^1$ forged by coalitions of at most t traitors can be traced using the above algorithm with success probability at least $1 - \epsilon_f$.

Note that this method works only for perfect decoders. We will give the complete method including the tracing of imperfect decoders in the full version of the paper.

Algorithm 1 $\texttt{Trace}(\mathcal{D}_S^1, PK, TK)$

Produce $\{W(j)\}_{j\in[n]} \leftarrow \mathcal{F}.\texttt{Gen}(1^n)$
Initialize a pirate codeword $w \leftarrow 0^\ell$
for $j \leftarrow 1$ to ℓ **do**
  $S_j = \{i : W(i)[j] = 1\}$
  $S_{j,1} \leftarrow S \cap S_j$
  $S_{j,2} \leftarrow S \setminus S_j$
  $(hdr_1, K_1) \leftarrow \texttt{BEnc}(PK, S_{j,1})$
  $(hdr_2, K_2) \leftarrow \texttt{BEnc}(PK, S_{j,2})$
  Choose k, r randomly
  $c_1 \leftarrow hdr_1 || SymEnc_{K_1}(k)$
  $c_2 \leftarrow hdr_2 || SymEnc_{K_2}(r)$
  $c \leftarrow c_1 || c_2$
  $k' \leftarrow \mathcal{D}_S^1(c)$
  if $k' = k$ **then**
    $w_j \leftarrow 1$
  else
    $w_j \leftarrow 0$
$t \leftarrow \mathcal{F}.\texttt{Identify}(w)$ and accuse u_t

4 Comparison with Existing T&R Schemes

In this section, we will discuss the performance of the resulting T&R schemes that emerges when we use some of the popular BE schemes in the literature with our construction technique. The comparison is summarized in Table 1.

Trace&Revoke Schemes	Public Key Size	Private Key Size	Ciphertext Size	Security & Type
BW[3]	$O(\sqrt{n})$	$O(\sqrt{n})$	$O(\sqrt{n})$	Adaptive
FA[7]	$O(n)$	$O(1)$	$O(\sqrt{n})$	Ad/Generic GM
Our Results				
$\texttt{T\&R-BGW}_1$	$O(n)$	$O(1)$	$O(1)$	Static
$\texttt{T\&R-BGW}_2$	$O(\sqrt{n})$	$O(1)$	$O(\sqrt{n})$	Static
$\texttt{T\&R-Del}_1$	$O(n)$	$O(1)$	$O(1)$	Static/ID-based
$\texttt{T\&R-Del}_2$	$O(\sqrt{n})$	$O(1)$	$O(\sqrt{n})$	Static/ID-based
$\texttt{T\&R-GW}_1$	$O(m)$	$O(1)$	$O(1)$	Static
$\texttt{T\&R-GW}_2$	$O(n)$	$O(1)$	$O(1)$	Ad/ROM/ID-based
$\texttt{T\&R-GW}_3$	$O(\sqrt{n})$	$O(1)$	$O(\sqrt{n})$	Ad/ID-based

Table 1: Our construction applied to the BE schemes that are proposed by [1, 5, 8] and the results are compared with the T&R scheme of [3] and [7]. Note that different schemes in these papers are distinguished by indices. The value m in $\texttt{GW}_1$ is the maximum number of recipients in a single broadcast.

Particularly, we instantiated our method with three different BE schemes which we will call BGW [1], Del [5] (identical to SF [11]), and GW [8]. We compared the results with the only two known T&R schemes(to the best of our knowledge), namely BW [3] and FA [7].

BW and FA has $O(\sqrt{n})$ size ciphertexts while they offer a trade-off between public key and private key sizes, their product being $O(n)$. Note that the schemes generated by our method with similar security measures beat their complexities.

Round Complexity Round complexity is the number of queries needed in tracing thus directly affects the time complexity of tracing. Our construction which uses [1] and Tardos

codes [12], has a round complexity of $O(\frac{n^2}{\delta^2}\log\frac{n}{\epsilon}\log\frac{n^2\log\frac{n}{\epsilon}}{\epsilon})$ with an appropriate selection of parameters. This complexity is similar to that of [3] which is $O(\frac{n^2\log n}{\sigma^2}\log\frac{1}{\epsilon})$. However, an interesting point of our method is that it allows partially collusion secure constructions and in such a construction, the round complexity of tracing becomes quadratic in terms of the collusion bound which we call w instead of n.

References

[1] D. Boneh, C. Gentry, and B. Waters. Collusion resistent broadcast encryption with shorter ciphertexts and private keys. In *CRYPTO'05*, volume 3621 of *LNCS*, pages 258–275. Springer-Verlag, 2005.

[2] D. Boneh and J. Shaw, Collusion-Secure Fingerprinting for Digital Data, IEEE Transactions on Information Theory, Vol. 44(5) pp. 1897-1905, 1998.

[3] D. Boneh and B. Waters. A fully collusion resistent broadcast, trace, and revoke system. In *CCS '06*, pages 211–220. ACM, 2006.

[4] B. Chor, A. Fiat, and M. Naor, Tracing Traitors, CRYPTO '94, LNCS 839 Springer 1994, pp. 257-270.

[5] C. Delerableé. Identity-Based Broadcast Encryption with Constant Size Ciphertexts and Private Keys. In *ASIACRYPT'07*, volume 4833 of *LNCS*, pages 200–215. Springer-Verlag, 2008.

[6] A. Fiat and M. Naor. Broadcast encryption. In *CRYPTO'93*, volume 773 of *LNCS*, pages 480–491. Springer-Verlag, 1993.

[7] J. Furukawa and N. Attrapadung. Fully Collusion Resistant Black-Box Traitor Revocable Broadcast Encryption with Short Private Keys. In *Automata, Languages and Programming*, volume 4596 of *LNCS*, pages 496–508. Springer-Verlag, 2007.

[8] C. Gentry, B. Waters. Adaptive Security in Broadcast Encryption Systems (with Short Ciphertexts). In EUROCRYPT '09, Lecture Notes in Computer Science, 2009, Volume 5479/2009, 171-188.

[9] A. Kiayias, S. Pehlivanoglu. Improving the Round Complexity of Traitor Tracing Schemes. In *Applied Cryptography and Network Security*, vol. 6123, pages 273–290, 2010.

[10] A. Kiayias, S. Pehlivanoglu. Encryption for Digital Content, vol. 52 of Advances in Information Security. Springer US, 2010, ISBN: 10.1007/978-1-4419-0044-9.

[11] R. Sakai, J. Furukawa. Identity-Based Broadcast Encryption. In *Cryptology ePrint Archive*, Report 2007/21.

[12] G. Tardos. Optimal probabilistic fingerprint codes. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, STOC '03, pages 116–125, 2003.