Threshold Traitor Tracing with Public Traceability

Selected Areas in Cryptography 2026

Sébastien Canard, Nathan Papon and Duong Hieu Phan
nathan.papon@telecom-paris.fr
August 26th 2026

Plan

Plan

Threshold Cryptography [DF90]

Given $ct = Enc(pk, pt)$:

- Public Key Encryption :

$$Dec(sk, ct) = pt$$

- Threshold Decryption :

$$\cancel{Dec(sk, ct) = pt} \implies PartDec(sk_i, ct) = pd_i$$

- Then combine the shares to recover pt :

$$Combine(pk, \{pd_i\}_{i \in T}) = pt$$

Security guarantee of a threshold scheme

With less than $t - 1$ corruptions, the adversary learns nothing on pt.

What happens with more than t corruptions ?

It seems that there is no security ... is there ?

Plan

Threshold Traitor Tracing [BPR24]

- Extends Threshold Decryption
- Suppose the adversary corrupts more that t parties and creates a decoder D [1].
- The goal is to find at least one traitor from the coalition.

1. In [BPR24], it is shown that with less than t traitors tracing is almost impossible.

The Decoder

- Suppose that the traitors craft an obfuscated decoder D.
- D acts as a black-box.
- D has no memory.
- Given a ciphertext ct, D outputs a plaintext pt with non negligble probability [2] :

$$\Pr\left[D(ct) = pt\right] = \varepsilon$$

2. More precisely, the decoder distinguishes between encryption of two plaintext with non negligible probability

Our work

$$Trace(pk, tk, D, ct) = \mathcal{J}$$

- Private Tracing : tk is owned by a *tracing authority*.
- Public Tracing : tk is public or $\perp$, everybody can trace.

Strengthens a framework in which *a single party should not possess too much power*.

This work

Construct a Threshold Traitor Tracing Scheme that supports public tracing.

Plan

Plan

Traceable codes
(Informal definition)

Let $\mathcal{C} = \{\omega_1, \ldots, \omega_N\}$ be a Q-ary code of size N and length ℓ.
Let $\omega^i = \omega_1^i \ldots \omega_\ell^i$ and $\omega^j = \omega_1^j \ldots \omega_\ell^j$ be two codewords of $\mathcal{C}$.
A traceable code has a *Trace* algorithm such that :

$$Trace(tk, \omega^b = \omega_1^{b_1} \ldots \omega_\ell^{b_\ell}) \in \{\omega^i, \omega^j\}.$$

We use codes with Identifiable Parent Property (IPP) [BKM10] : $tk = \perp$, tracing is public.

A simple example

Let $\Sigma = [a, b, \ldots, z]$, $\omega^* = nadiance$. Then, $Trace(tk, \omega^*) = \begin{cases} nathan \\ audience \end{cases}$.

nathan and *audience* are *parents* codewords of *nadiance*.

Plan

Q-Partite Threshold Encryption

- $Setup(1^\lambda, N, \ell, t, Q) \to (pk, \{sk_i\}_{i\in[N]})$ with

$$pk = \begin{pmatrix} pk_{1,1} & \cdots & pk_{1,Q} \\ & \vdots & \\ pk_{j,1} & \cdots & pk_{j,Q} \\ & \vdots & \\ pk_{\ell,1} & \cdots & pk_{\ell,Q} \end{pmatrix}, \; sk_i = \begin{pmatrix} sk_{i,1}^1 & \cdots & sk_{i,\alpha}^1 & \cdots & sk_{i,Q}^1 \\ & & \vdots & & \\ sk_{i,1}^j & \cdots & sk_{i,\alpha}^j & \cdots & sk_{i,Q}^j \\ & & \vdots & & \\ sk_{i,1}^\ell & \cdots & sk_{i,\alpha}^\ell & \cdots & sk_{i,Q}^\ell \end{pmatrix}$$

- $Enc(pk, j, pt) \to ct = (ct_0, ct_1, \ldots, ct_\alpha, \ldots, ct_Q)$
- $PartDec(sk_i, ct) \to pd_i$
- $Combine(pk, \{pd_i\}) \to pt$

Security of the Q-Partite PKE

On top of *CPA*-style security (for threshold schemes), two notions are essential to this scheme :

- *all-sided correctness* : Every pair $(ct_\alpha, sk^j_{i,\alpha})$ output the same share pd_i.
- *(adaptive) One Sided Security* : $ct_{\alpha^*} \approx rand$ if $\mathcal{A}$ has no $sk^{j^*}_{i,\alpha^*}$

Plan

Combining the IPP code with Q-Partite TPKE

- IPP codes trace the traitor.
- Q-Partite TPKE encrypts and decrypts messages.

We can combine them to construct a *Threshold Traitor Tracing scheme* with public tracing.

Combining the IPP code with Q-Partite TPKE

Threshold Traitor Tracing

- $Setup(1^\lambda, N, \ell, t, Q)$: For each $i \in [N]$, sample ω_i from IPP code $\mathcal{C}$ and set $sk_i = (sk^1_{i,\omega^i_1}, \cdots, sk^\ell_{i,\omega^i_\ell})$.
- Enc, $PartDec$, $Combine$ are the same.

Goal : construct a word $\omega^* = \omega_1^* \dots \omega_\ell^*$. To set ω_j^*, repeat the following :

- *Step* 1 : Use D on input $ct = (ct_0, ct_1, \dots, ct_Q) = Enc(pk, j, pt)$.
- Let p_1 be the probability that D answers correctly.
- p_1 is *high*, since ct is a real ciphertext.

Goal : construct a word $\omega^* = \omega_1^* \dots \omega_\ell^*$. To set ω_j^*, repeat the following :

- *Step* 2 : Use D decoder on input $ct = (ct_0, ct_1, \dots, ct_{Q-1}, ct_Q)$.
- Let p_2 be the probability that D answers correctly.

Tracing Traitors
The tracing

Goal : construct a word $\omega^* = \omega_1^* \dots \omega_\ell^*$. To set ω_j^*, repeat the following :

- *Step* α : Use D decoder on input $ct = (ct_0, ct_1, \dots, ct_{\alpha-1}, ct_\alpha, \dots, ct_Q)$
- Let p_α be the probability that D answers correctly.

Goal : construct a word $\omega^* = \omega_1^* \dots \omega_\ell^*$. To set ω_j^*, repeat the following :

- *Step Q* : Use D decoder on input $ct = (ct_0, ct_1, \dots, ct_Q)$
- Let p_Q be the probability that D answers correctly.
- p_Q is *low*, since ct is entirely random.

Goal : construct a word $\omega^* = \omega_1^* \dots \omega_\ell^*$. Subgoal : set ω_j^*.
Since p_1 is high and p_Q is low, then there must be a letter $\alpha \in [Q]$ such that $p_{\alpha-1} - p_\alpha$ is significant ($\geq Adv_D/\#repetitions$).

Plan

The Learning With Errors Problem

Definition (Decisional LWE)

Let n,q,m be positive integers. Let $\mathcal{D}_{\mathbb{Z}_q,\sigma}$ be the discrete gaussian distribution over $\mathbb{Z}_q$ with width σ. The decision LWE problem for (n, m, q, σ) is to distinguish between the following to distributions :

$$\left\{(A, b = A^\top s + e) \ : \ A \leftarrow \mathbb{Z}_q^{n\times m}, s \leftarrow \mathbb{Z}_q^n, e \leftarrow \mathcal{D}_{\mathbb{Z}_q,\sigma}^m\right\}$$
$$\left\{(A, u) \ : \ A \leftarrow \mathbb{Z}_q^{n\times m}, u \leftarrow \mathbb{Z}_q^m\right\}$$

Starting point : Regev's Encryption

- Public key : $pk = (A, b = A^\top s + e) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^m$, Secret key : $sk = \in \mathbb{Z}_q^n$.
- Encryption : Sample $r \leftarrow \{0,1\}^m$, create ciphertext $ct = (ct_0, ct_1)$ as $(ct_0, ct_1) = \left(Ar, b^\top r + \mu \lfloor \frac{q}{2} \rceil\right)$
- Decryption : Compute $ct_1 - sk^\top ct_0$ and round.

Thresholdising Regev's Encryption

- Use a secret sharing on sk : $x = s_1 + s_2 + \ldots + s_N$
- Encryption is the same : $ct = (ct_0, ct_1) = \left(Ar, b^\top r + \mu \lfloor \frac{q}{2} \rceil\right)$
- Partial Decryption : $pd_i = sk_i^\top ct_0$
- Combine : Compute $ct_1 - \sum_{i \in [N]} pd_i$ and round.

Extending the keys : key shifting

- Sample $x, y_1, \ldots, y_Q \leftarrow \mathbb{Z}_q^n$ together with $e, e_1, \ldots, e_Q \leftarrow \mathcal{D}_{\mathbb{Z}_q,\sigma}^m$.
- Set the public key pk as $pk = (A^\top x + e, A^\top y_1 + e_1, \ldots, A^\top y_Q + e_Q)$.
- Secret share x as $x = \sum_{i\in[N]} s_i \in \mathbb{Z}_q^n$.
- Set the secret keys as sk_i as $sk_{i,1} = s_i - y_1, \ldots, sk_{i,Q} = s_i - y_Q$.

Encryption and Decryption under shifted keys

- Encryption : Sample $r \leftarrow \{0,1\}^m$, compute $ct = (ct_r, ct_0, \ldots, ct_1)$ as :
  - $ct_r = Ar \in \mathbb{Z}_q^m$
  - $ct_0 = x^\top Ar + e^\top r + \mu\lfloor \frac{q}{2} \rceil \in \mathbb{Z}_q$
  - $ct_1 = y_1^\top Ar + e_1^\top r, \ldots, ct_Q = y_Q^\top Ar + e_Q^\top r \in \mathbb{Z}_q$
- Decryption : Chose $\alpha \in [Q]$ and compute $pd_i = sk_{i,\alpha}^\top ct_r + ct_\alpha$
- Combine : Compute $ct_0 - \sum_{i\in[N]} pd_i$ and round.

Security

- *CPA security* : Given at most $N-1$ secret keys, argue that $(A, A^\top x + e) \approx (A, u) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^m$.
- *One sided security* : Given all keys except $\{sk_{i,\alpha^*}\}_{i \in [N]}$ for some $\alpha^* \in [Q]$, argue that $(A, A^\top y_{\alpha^*} + e_{\alpha^*}) \approx (A, u) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^m$.

Plan

Conclusion and main takeaway

- Combined with codes with identifiable parents property, this gives a Threshold Traitor Tracing where tracing is *public*.
- Construction proven secure under standard LWE assumption.

Future Works

- Reduce the modulus size q (ideally to polynomial size).
- Improve the secret sharing scheme.


Dan Boneh, Aggelos Kiayias, and Hart William Montgomery.
Robust fingerprinting codes : a near optimal construction.
In *Proceedings of the Tenth Annual ACM Workshop on Digital Rights Management*, DRM '10, page 3–12, New York, NY, USA, 2010. Association for Computing Machinery.


Dan Boneh, Aditi Partap, and Lior Rotem.
Accountability for misbehavior in threshold decryption via threshold traitor tracing.
pages 317–351, 2024.


Yvo Desmedt and Yair Frankel.
Threshold cryptosystems.
pages 307–315, 1990.

Construction of the Q-TPKE

Let $\mathbb{G}$ be a cyclic group of order q and generator g.

- $Setup(1^\lambda, N, \ell, t, Q)$: For each position $j \in [\ell]$, sample $x_j, y_{j,1}, \ldots, y_{j,Q} \leftarrow \mathbb{Z}_q$. Set $pk_j = (X_j, Y_{j,1}, \ldots, Y_{j,Q})$ with $X_j = g^{x_j}$, $Y_{j,1} = g^{y_{j,1}}, \ldots, Y_{j,Q} = g^{y_{j,Q}}$. Let $s_{j,1}, \ldots, s_{j,N} \leftarrow Shamir(t, N, x_j)$. Set $sk^j_{i,1} = \frac{s_{j,1}}{y_{j,1}}, \ldots, sk^j_{i,Q} = \frac{s_{j,Q}}{y_{j,Q}}$.
- $Encrypt(pk, j, M \in \mathbb{G})$: Sample $r \leftarrow \mathbb{Z}_q$, set $ct = (ct_0, ct_1, \ldots, ct_Q)$ with $ct_0 = MX_j^r, ct_1 = Y_{j,1}^r, \ldots, ct_Q = Y_{j,Q}^r$.
- $PartDec(pk, sk^j{}_{,i,\alpha}, ct)$: Compute $d_i = ct_\alpha^{sk^j{}_{,i,\alpha}} = g^{rs_{j,i}}$.
- $Combine(pk, ct, \{d_i\}_{i \in J})$: Compute Lagrange coefficients λ_i^J associated with set J, then compute $M = ct_0 / \Pi d_i^{\lambda_i^J}$.

Collusion Resistant Fingerprinting Codes

Noisy codewords are ternary words $\omega \in \{0,1,?\}^\ell$. For a set $W = (\omega^{(1)}, \ldots, \omega^{(k)})$, a word $\omega^\star$ is *feasible* if for each $j \in [\ell]$, there is $i[k]$ s.t $\omega^\star_j = \omega^{(i)}_j$ or $\omega^\star = ?$. We denote by $\delta - F(W)$ the set of *feasible* words for W with at most $\delta\ell$? symbols.

A fingerprinting code is a pair of algorithms $(Gen, Trace)$ such that :

- $Gen(1^N, \varepsilon, \delta)$: Outputs a code $\mathcal{C} \subseteq \{0,1\}^\ell$ and a tracing key tk. The length ℓ is a function of the size n, security parameter ε and noise bound δ.
- $Trace(tk, \omega^\star)$: Outputs a subset $S \subseteq [N]$.

Codes with Robust Identifying Parents Property

Given a code $\mathcal{C} \subseteq \Sigma^{\ell}$ with parameters (ℓ, N, Q), let $t \geq 2$, be an integer. The code $\mathcal{C}$ is called a (t, ε)-*IPP* (robust t-*IPP*) if it holds that

$$\bigcap_{X \subset \mathcal{C}, |X| \leq t, x \in desc(X)_{\varepsilon}} X \neq \emptyset$$

Parameters comparison

- For Fingerprinting Codes, $\ell = O(f^2 \log n^2 \log(N/\kappa)/(1-\theta))$ where N is the number of users, f is the maximal number of traitors, κ the error probability in identifying traitors and θ the maximal proportion of letters deviating from the marking assumption.
- For IPP Codes, $\ell = O(\frac{2d \log n}{\log(d \log n)})$ where $d = f^2/(1-\theta-f\theta)$ and alphabet size $q = \ell$.

$\Rightarrow$ IPP codewords are shorter *but* alphabet size is a larger.