Secure Codes with List Decoding

顧 玉杰 (GU Yujie)

九州大学
KYUSHU UNIVERSITY

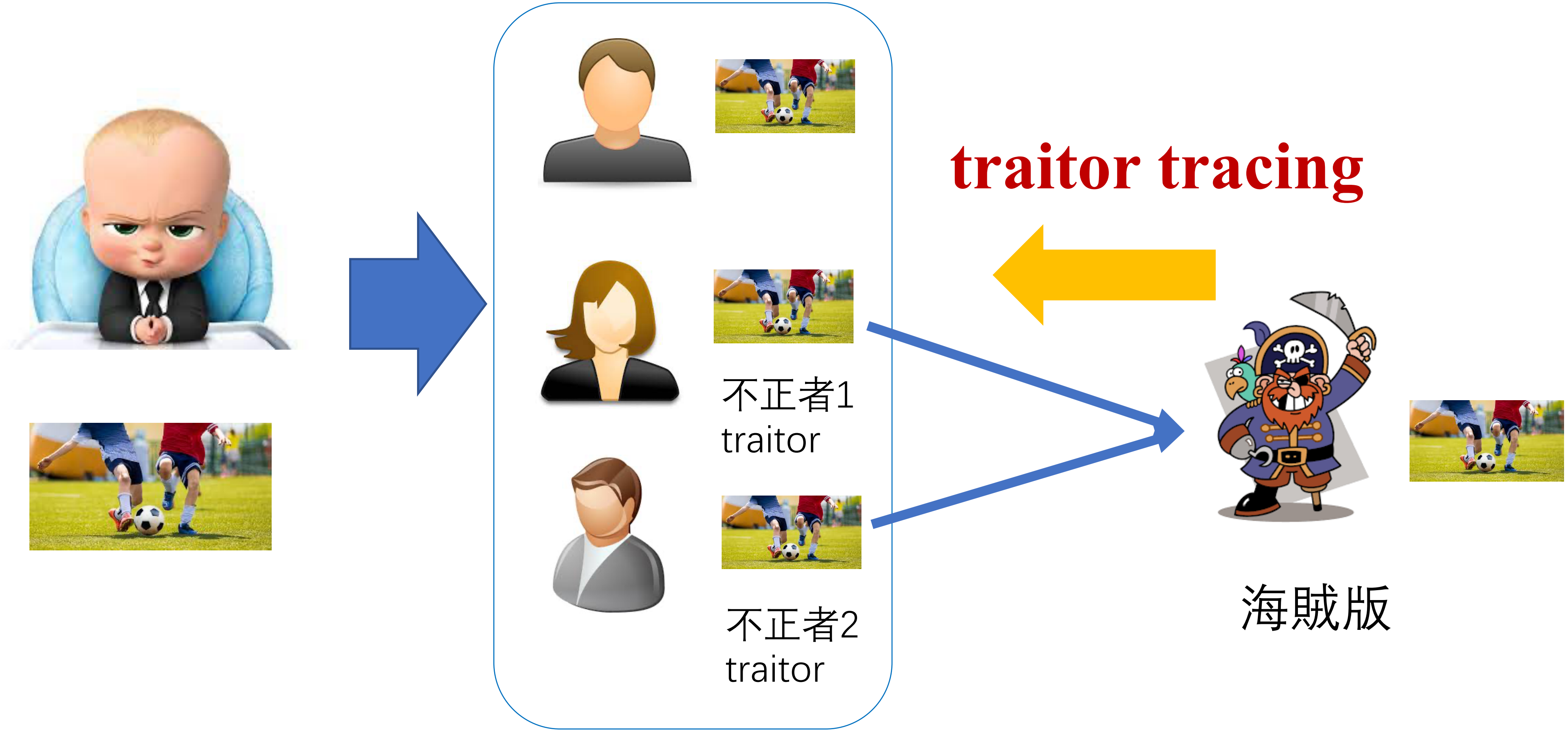
2022.05.17 @ Gifu

Secure Codes for ?

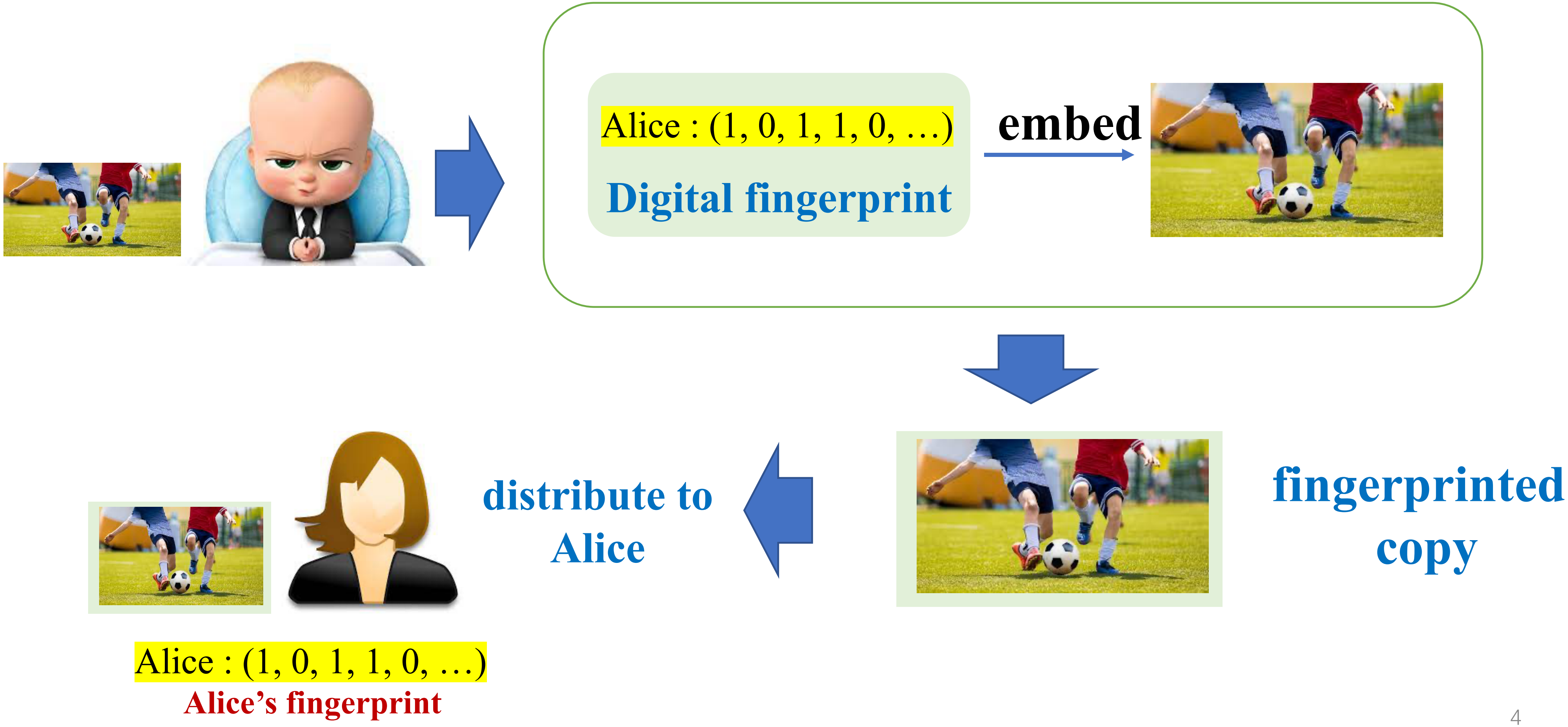
(**multimedia content:** text, audio, images, animations, video…)

Traitor Tracing

- Chor-Fiat-Naor, CRYPTO-1994



Embed fingerprints



Collusion attack

collusion attack

Alice

Alice : (1, 0, 1, 1, 0, …)

Bob

Bob : (0, 1, 1, 0, 0, …)

(0.5, 0.7, 1, 0.2, 0, …)

colluded (pirate) copy

different users – different fingerprints

Traitor Tracing

pirate copy

(0.5, 0.7, 1, 0.2, 0, …)

extract fingerprint

Secure (fingerprinting) codes

Identify traitors (decoding algorithm)

Alice, Bob, …

研究課題：Design **Secure (fingerprinting) Codes:**

- **large code rate + efficient identifying algorithm**

Code Modulation

- host signal: a real vector $\mathbf{x} \in \mathbb{R}^n$
- n orthogonal basis signals $\{\mathbf{u}_i \in \mathbb{R}^v : 1 \leq i \leq n\}$
- a *fingerprint* :

$$\mathbf{w}_j = \sum_{i=1}^{n} \mathbf{c}_j(i)\mathbf{u}_i$$

  where $\mathbf{c}_j(i) \in \{0, 1\}$.

- user j: a fingerprinted signal copy of $\mathbf{x}$:

$$\mathbf{y}_j = \mathbf{x} + \mathbf{w}_j$$

- user $j \iff$ fingerprint $\mathbf{w}_j \iff$ the vector

$$\mathbf{c}_j = (\mathbf{c}_j(1), \ldots, \mathbf{c}_j(n)) \in \{0, 1\}^n$$

- all M authorized users $\iff$ fingerprinting code

$$\{\mathbf{c}_j \in \{0, 1\}^n : 1 \leq j \leq M\}$$

Linear Collusion Attack

- a coalition: $J \subseteq \{1, \ldots, M\}$ malicious users
- a forged copy

$$\hat{\mathbf{y}} = \sum_{j \in J} \lambda_j \mathbf{y}_j = \sum_{j \in J} \lambda_j (\mathbf{x} + \mathbf{w}_j) = \mathbf{x} + \sum_{j \in J} \lambda_j \mathbf{w}_j$$

where $\lambda_j \in \mathbb{R}$ such that $0 < \lambda_j < 1$ and

$$\sum_{j \in J} \lambda_j = 1$$

Analysis

$$\begin{aligned}\hat{\mathbf{y}} &= \mathbf{x} + \sum_{j \in J} \lambda_j \mathbf{w}_j \\ &= \mathbf{x} + \sum_{j \in J} \lambda_j \sum_{i=1}^{n} \mathbf{c}_j(i) \mathbf{u}_i \\ &= \mathbf{x} + \sum_{i=1}^{n} \left(\sum_{j \in J} \lambda_j \mathbf{c}_j(i) \right) \mathbf{u}_i\end{aligned}$$

- $\langle \hat{\mathbf{y}} - \mathbf{x}, \mathbf{u}_i \rangle = \sum_{j \in J} \lambda_j \mathbf{c}_j(i)$
- If $\sum_{j \in J} \lambda_j \mathbf{c}_j(i) = 0$, then
$$\{\mathbf{c}_j(i) : j \in J\} = \{0\}.$$
- If $\sum_{j \in J} \lambda_j \mathbf{c}_j(i) = 1$, then
$$\{\mathbf{c}_j(i) : j \in J\} = \{1\}.$$
- If $0 < \sum_{j \in J} \lambda_j \mathbf{c}_j(i) < 1$, then
$$\{\mathbf{c}_j(i) : j \in J\} = \{0, 1\}.$$

Codes

- $Q = \{0,1,..,q-1\}$.
- (n, M, q) code: $C \subseteq Q^n, |C| = \mathrm{M}$
- $P(Q)$ is the power set of Q.

For code $\mathcal{C} \subseteq Q^n$ we define its ith coordinates set as

$$\mathcal{C}(i) = \{\mathbf{c}(i) \in Q : \mathbf{c} \in \mathcal{C}\},\ 1 \le i \le n.$$

The *descendant code* of $\mathcal{C}$ is defined as

$$\mathrm{desc}(\mathcal{C}) = (\mathcal{C}(1), \cdots, \mathcal{C}(n)) \in \mathcal{P}(Q)^n$$

- 例： if $\mathcal{C} = \{000, 011, 012\}$, then $\mathrm{desc}(\mathcal{C}) = (\{0\}, \{0,1\}, \{0,1,2\})$

- $a \in Q^n,\ b \in P(Q)^n$

we define a *partial order* $\mathbf{a} \preceq \mathbf{b}$ by the inclusion, that is, $\mathbf{a} \preceq \mathbf{b}$ if and only if $\mathbf{a}(i) \in \mathbf{b}(i)$ for all $i \in [n]$.

例: $(0,0,0) \preccurlyeq (\{0\},\{0,1\},\{0,1,2\})$

Secure codes

- (n, M, q) code: $C = \{c_1, c_2, \ldots, c_M\} \subseteq Q^n$
- $\exists$ an identifying/decoding algorithm $\mathcal{A}$ such that
- $\forall$ coalition $C_0 \subseteq C$ such that $|C_0| \leq t$
  and the pirate $\mathrm{d} = \mathrm{desc}(C_0)$
- we have

$$\mathcal{A}(\mathrm{d}) = C_0 \qquad \text{(complete traceability)}$$

$$\emptyset \neq \mathcal{A}(\mathrm{d}) \subseteq C_0 \qquad \text{(partial traceability)}$$

To guarantee $\mathcal{A}$, we need some requirements on C

Frameproof codes

- **Boneh and Shaw, 1998, IEEE-TIT**

- ***t-Frameproof Code***, $t\text{-}FPC(n, M, q)$
- $C = \{c_1, c_2, \ldots, c_M\} \subseteq Q^n$
- $\forall\ C_0 \subseteq C$ s.t. $|C_0| \leq t$
- $\forall\ a \in C¥C_0$
- we have

$$a \not\preccurlyeq \mathrm{desc}(C_0)$$

例：decoding of frameproof codes

# of traitors = 2

$$\begin{pmatrix} 0 & 0 & 1 & 1 & 2 \\ 0 & 1 & 2 & 1 & 1 \\ 0 & 2 & 0 & 1 & 2 \end{pmatrix} \xrightarrow{\begin{pmatrix} \{0,1\} \\ \{1,2\} \\ \{0,2\} \end{pmatrix}} \begin{pmatrix} 0 & 0 & 1 & 1 & 2 \\ 0 & 1 & 2 & 1 & 1 \\ 0 & 2 & 0 & 1 & 2 \end{pmatrix}$$

traitors

Remove all columns c such that c $\npreceq \begin{pmatrix} \{0,1\} \\ \{1,2\} \\ \{0,2\} \end{pmatrix}$

· The decoding complexity is $O(nM)$.

Separable codes

- **Cheng and Miao, 2011, IEEE-TIT**

> - ***t-Separable Code***, t-$SC(n, M, q)$
> - $C = \{c_1, c_2, \ldots, c_M\} \subseteq Q^n$
> - $\forall\ C_1 \subseteq C, C_2 \subseteq C$ s.t. $|C_1| \leq t, |C_2| \leq t,\ C_1 \neq C_2$
> - we have
>
> $$\mathrm{desc}(C_1) \neq \mathrm{desc}(C_2)$$

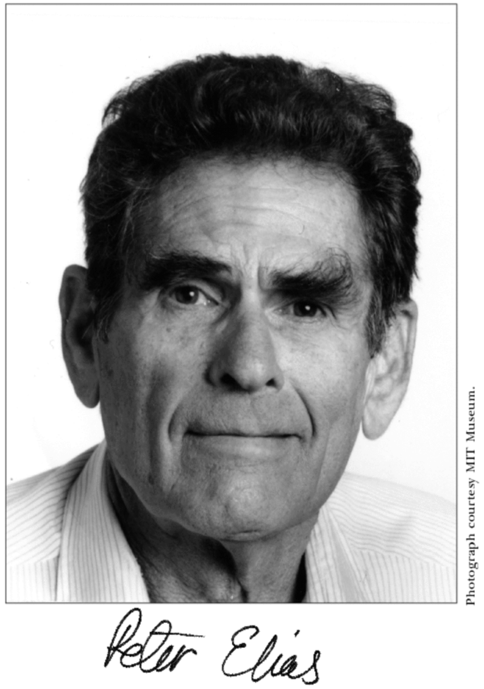
- The decoding complexity is $O(nM^t)$. (check all t-subsets)

Look Closer

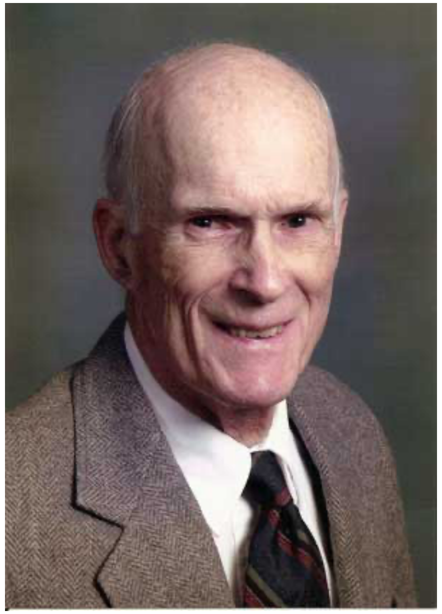
	frameproof code	separable code
decoding cost	$O(nM)$	$O(nM^t)$
code rate (number of users)	R_{FP} <	R_{SC}

Motivation: low cost + high code rate

List Decoding (error-correcting codes)



Peter Elias



John Wozencraft

(1950s)

Secure Codes with List Decoding

- **G.-Vorobyev-Miao, 2022**

- ***t-Secure Code with List Decoding***, $t\text{-}SCLD(n, M, q; L)$
- $C = \{c_1, c_2, \dots, c_M\} \subseteq Q^n$
- $\forall\, C_1 \subseteq C, C_2 \subseteq C$ s.t. $|C_1| \le t, |C_2| \le t,\ C_1 \neq C_2$
- $\mathrm{d} = \mathrm{desc}(C_1)$
- we have

(1) $|\mathrm{Res}(\mathrm{d})| = |\{c \in C:\ c \preccurlyeq \mathrm{d}\}| \le L$

(2) $\mathrm{desc}(C_1) \neq \mathrm{desc}(C_2)$

A unified concept

- Many existing secure codes can be seen as special cases of SCLD (by taking specific L).
- For example:
  - A t-$FPC(n, M, q)$ is a t-$SCLD\ (n, M, q; L = t)$.
  - A t-$SC(n, M, q)$ is a t-$SCLD\ (n, M, q; L = M)$.

Decoding of SCLD

Algorithm 1 Identifying algorithm for $\bar{t}$-SCLD

Input: a $\bar{t}$-SCLD $\mathcal{C}$; a vector $\mathbf{d} \in \mathrm{Desc}_t(\mathcal{C})$
Output: the set of all traitors $\mathcal{T}$

```
T = ∅, W = ∅.                                    ▷ Initialize the candidate sets
for each c ∈ C do                                ▷ Step 1
    if c ⪯ d then
        W = W ∪ {c};
    end if
end for
for each subset S ⊆ W of size at most t do       ▷ Step 2
    if desc(S) == d then
        T = S;
        output T;
    end if
end for
```

The decoding complexity is $\boldsymbol{O}(\max\{\boldsymbol{nM}, \boldsymbol{nL^t}\})$.

e.g. If $\boldsymbol{L} = \boldsymbol{M}^{1/t}$, then $\boldsymbol{O}(\boldsymbol{nM})$.

Use the Property (1) of SCLD:

- find a list W, $|W| \le L$
- cost $O(nM)$

Use the Property (2) of SCLD:

- find all exact traitors
- cost $O(nL^t)$

例：decoding of SCLD

# of traitors = 2

$$\begin{pmatrix} 0 & 0 & 1 & 1 & 2 & 2 \\ 0 & 1 & 2 & 1 & 1 & 2 \\ 0 & 2 & 0 & 2 & 2 & 2 \end{pmatrix}$$

$$\begin{pmatrix} \{0,1\} \\ \{1,2\} \\ \{0,2\} \end{pmatrix}$$

Step 1

Find a small list

Remove all columns c such that $c \npreceq \begin{pmatrix} \{0,1\} \\ \{1,2\} \\ \{0,2\} \end{pmatrix}$

$$\begin{pmatrix} 0 & 0 & 1 & 1 & 2 & 2 \\ 0 & 1 & 2 & 1 & 1 & 2 \\ 0 & 2 & 0 & 2 & 2 & 2 \end{pmatrix}$$

Step 2

Check all 2-subsets in the list

$$\begin{pmatrix} 0 & 0 & 1 & 1 & 2 & 2 \\ 0 & 1 & 2 & 1 & 1 & 2 \\ 0 & 2 & 0 & 2 & 2 & 2 \end{pmatrix}$$

traitors

Code Rate of SCLD

- State-of-the-art lower bounds (new results in bold)

	t	2	3	4	5	6
$O(nM^{t})$	$R_{\mathrm{SC}}(t) \geq$	0.5	**0.13834**	**0.06198**	**0.03138**	**0.02003**
$O(nM)$	$R^{(1/t)}_{\mathrm{SCLD}}(t) \geq$	**0.44452**	**0.13205**	**0.05770**	**0.03105**	**0.01997**
$O(nM)$	$R_{\mathrm{FPC}}(t) \geq$	0.20756	0.07999	0.04392	0.02794	0.01936

Proof：random coding

Part of the results have been accepted to IEEE ISIT-2022.
The full paper is coming soon.

Dynamic Traitor Tracing

- **Fiat-Tassa, CRYPTO-1999**

dynamic decoding ← dynamic encoding

Merit: accommodate more users

Dynamic Traitor Tracing

Algorithm 2 Two-stage dynamic traitor tracing algorithm

Input: t-HLD code $\mathcal{C} = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_M\}$ with list size $L^{(1)} = \Theta(M^{\alpha})$; the pirate $\mathbf{d}^{(1)} \in \mathcal{P}(Q)^n$ ▷ Stage 1

1: $\mathcal{T} = \emptyset$, $\mathcal{W} = \emptyset$. ▷ Initialize the candidate sets

2: **for** each $j \in [M]$ **do**

3: **if** $\mathbf{c}_j \preceq \mathbf{d}^{(1)}$ **then**

4: $\mathcal{W} = \mathcal{W} \cup \{j\}$;

5: **end if**

6: **end for**

Output: the index set $\mathcal{W}$

Stage 1: Only use Property (1) of SCLD

Input: t-SCLD code $\mathcal{C} = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_{|\mathcal{W}|}\}$ with list size $L^{(2)} = \Theta(|\mathcal{W}|^{\beta})$; the pirate $\mathbf{d}^{(2)} \in \mathcal{P}(Q)^n$ ▷ Stage 2

1: execute the two steps as in Algorithm 1 for SCLD

Output: the set of all traitors $\mathcal{T}$

Stage 2: Use decoding of SCLD

- **Code Rate : two-stage dynamic traitor tracing > SCLD**

Look for more

- Explicit construction of SCLD
  - algebraic property → decoding in time $O(\mathbf{polylog}(nM))$?
- More application of list decoding (and beyond)
  - various applications + list decoding

Thank organizers for the invitation.

Thank you all for the listening.