TU/e Technische Universiteit Eindhoven University of Technology

Dealing with Specific Pirate Attacks in Collusion-Resistant Traitor Tracing

Thijs Laarhoven

mail@thijs.com
http://www.thijs.com/

Guangzhou, China
(November 20, 2013)

TU/e

Outline

Collusion-resistant traitor tracing

Score-based construction

Fighting against specific attacks

Results

Conclusion

TU/e

Collusion-resistant traitor tracing

Illegal redistribution

User	Copyrighted content															
Antonino	0	1	1	1	0	0	1	1	1	0	1	1	0	0	1	0	...
Boris	0	1	1	1	0	0	1	1	1	0	1	1	0	0	1	0	...
Caroline	0	1	1	1	0	0	1	1	1	0	1	1	0	0	1	0	...
David	0	1	1	1	0	0	1	1	1	0	1	1	0	0	1	0	...
Eve	0	1	1	1	0	0	1	1	1	0	1	1	0	0	1	0	...
Fred	0	1	1	1	0	0	1	1	1	0	1	1	0	0	1	0	...
Gábor	0	1	1	1	0	0	1	1	1	0	1	1	0	0	1	0	...
Henry	0	1	1	1	0	0	1	1	1	0	1	1	0	0	1	0	...

TU/e

Collusion-resistant traitor tracing

Illegal redistribution

User	Copyrighted content															
Antonino	0	1	1	1	0	0	1	1	1	0	1	1	0	0	1	0	...
Boris	0	1	1	1	0	0	1	1	1	0	1	1	0	0	1	0	...
Caroline	0	1	1	1	0	0	1	1	1	0	1	1	0	0	1	0	...
David	0	1	1	1	0	0	1	1	1	0	1	1	0	0	1	0	...
Eve	0	1	1	1	0	0	1	1	1	0	1	1	0	0	1	0	...
Fred	0	1	1	1	0	0	1	1	1	0	1	1	0	0	1	0	...
Gábor	0	1	1	1	0	0	1	1	1	0	1	1	0	0	1	0	...
Henry	0	1	1	1	0	0	1	1	1	0	1	1	0	0	1	0	...
Copy	0	1	1	1	0	0	1	1	1	0	1	1	0	0	1	0	...

TU/e

Collusion-resistant traitor tracing

Embedding fingerprints

User	Copyrighted content (fingerprinted)																
Antonino	0	1	1	1	0	0	1	1	1	0	1	1	0	1	0	0	...
Boris	0	1	1	1	0	1	0	1	1	0	1	1	1	1	1	0	...
Caroline	0	1	0	1	0	1	0	1	1	0	0	1	1	0	1	0	...
David	0	1	1	1	0	0	0	1	1	0	1	1	0	0	0	0	...
Eve	0	1	0	1	0	1	0	1	1	0	1	1	1	0	0	0	...
Fred	0	1	0	1	0	0	1	1	1	0	0	1	0	1	0	0	...
Gábor	0	1	1	1	0	1	1	1	1	0	1	1	0	0	1	0	...
Henry	0	1	0	1	0	1	1	1	1	0	0	1	0	1	1	0	...

TU/e

Collusion-resistant traitor tracing

Embedding fingerprints

User	Copyrighted content (fingerprinted)																
Antonino	0	1	1	1	0	0	1	1	1	0	1	1	0	1	0	0	...
Boris	0	1	1	1	0	1	0	1	1	0	1	1	1	1	1	0	...
Caroline	0	1	0	1	0	1	0	1	1	0	0	1	1	0	1	0	...
David	0	1	1	1	0	0	0	1	1	0	1	1	0	0	0	0	...
Eve	0	1	0	1	0	1	0	1	1	0	1	1	1	0	0	0	...
Fred	0	1	0	1	0	0	1	1	1	0	0	1	0	1	0	0	...
Gábor	0	1	1	1	0	1	1	1	1	0	1	1	0	0	1	0	...
Henry	0	1	0	1	0	1	1	1	1	0	0	1	0	1	1	0	...
Copy	0	1	0	1	0	1	0	1	1	0	1	1	1	0	0	0	...

TU/e

Collusion-resistant traitor tracing

Embedding fingerprints

User	Copyrighted content (fingerprinted)																
Antonino	0	1	**1**	1	0	**0**	**1**	1	1	0	**1**	1	**0**	**1**	**0**	0	...
Boris	0	1	**1**	1	0	**1**	**0**	1	1	0	**1**	1	**1**	**1**	**1**	0	...
Caroline	0	1	**0**	1	0	**1**	**0**	1	1	0	**0**	1	**1**	**0**	**1**	0	...
David	0	1	**1**	1	0	**0**	**0**	1	1	0	**1**	1	**0**	**0**	**0**	0	...
Eve	0	1	**0**	1	0	**1**	**0**	1	1	0	**1**	1	**1**	**0**	**0**	0	...
Fred	0	1	**0**	1	0	**0**	**1**	1	1	0	**0**	1	**0**	**1**	**0**	0	...
Gábor	0	1	**1**	1	0	**1**	**1**	1	1	0	**1**	1	**0**	**0**	**1**	0	...
Henry	0	1	**0**	1	0	**1**	**1**	1	1	0	**0**	1	**0**	**1**	**1**	0	...
Copy	0	1	**0**	1	0	**1**	**0**	1	1	0	**1**	1	**1**	**0**	**0**	0	...

TU/e

Collusion-resistant traitor tracing

Embedding fingerprints

User	Copyrighted content (fingerprinted)																
Antonino	0	1	**1**	1	0	**0**	**1**	1	1	0	**1**	1	**0**	**1**	**0**	0	...
Boris	0	1	**1**	1	0	**1**	**0**	1	1	0	**1**	1	**1**	**1**	**1**	0	...
Caroline	0	1	**0**	1	0	**1**	**0**	1	1	0	**0**	1	**1**	**0**	**1**	0	...
David	0	1	**1**	1	0	**0**	**0**	1	1	0	**1**	1	**0**	**0**	**0**	0	...
Eve	0	1	**0**	1	0	**1**	**0**	1	1	0	**1**	1	**1**	**0**	**0**	0	...
Fred	0	1	**0**	1	0	**0**	**1**	1	1	0	**0**	1	**0**	**1**	**0**	0	...
Gábor	0	1	**1**	1	0	**1**	**1**	1	1	0	**1**	1	**0**	**0**	**1**	0	...
Henry	0	1	**0**	1	0	**1**	**1**	1	1	0	**0**	1	**0**	**1**	**1**	0	...
Copy	0	1	**0**	1	0	**1**	**0**	1	1	0	**1**	1	**1**	**0**	**0**	0	...

TU/e

Collusion-resistant traitor tracing

Collusion attacks

User	Copyrighted content (fingerprinted)																
Antonino	0	1	1	1	0	0	1	1	1	0	1	1	0	1	0	0	...
Boris	0	1	1	1	0	1	0	1	1	0	1	1	1	1	1	0	...
Caroline	0	1	0	1	0	1	0	1	1	0	0	1	1	0	1	0	...
David	0	1	1	1	0	0	0	1	1	0	1	1	0	0	0	0	...
Eve	0	1	0	1	0	1	0	1	1	0	1	1	1	0	0	0	...
Fred	0	1	0	1	0	0	1	1	1	0	0	1	0	1	0	0	...
Gábor	0	1	1	1	0	1	1	1	1	0	1	1	0	0	1	0	...
Henry	0	1	0	1	0	1	1	1	1	0	0	1	0	1	1	0	...

TU/e

Collusion-resistant traitor tracing

Collusion attacks

User	Copyrighted content (fingerprinted)																
Antonino	0	1	1	1	0	0	1	1	1	0	1	1	0	1	0	0	...
Boris	0	1	1	1	0	1	0	1	1	0	1	1	1	1	1	0	...
Caroline	0	1	0	1	0	1	0	1	1	0	0	1	1	0	1	0	...
David	0	1	1	1	0	0	0	1	1	0	1	1	0	0	0	0	...
Eve	0	1	0	1	0	1	0	1	1	0	1	1	1	0	0	0	...
Fred	0	1	0	1	0	0	1	1	1	0	0	1	0	1	0	0	...
Gábor	0	1	1	1	0	1	1	1	1	0	1	1	0	0	1	0	...
Henry	0	1	0	1	0	1	1	1	1	0	0	1	0	1	1	0	...
Copy	0	1	1	1	0	1	0	1	1	0	1	1	0	1	0	0	...

TU/e

Collusion-resistant traitor tracing

Collusion attacks

User	Copyrighted content (fingerprinted)															
Antonino	0	1	**1**	1	0	**0**	**1**	1	1	0	**1**	1	**0**	**1**	**0**	0	...
Boris	0	1	**1**	1	0	**1**	**0**	1	1	0	**1**	1	**1**	**1**	**1**	0	...
Caroline	0	1	**0**	1	0	**1**	**0**	1	1	0	**0**	1	**1**	**0**	**1**	0	...
David	0	1	**1**	1	0	**0**	**0**	1	1	0	**1**	1	**0**	**0**	**0**	0	...
Eve	0	1	**0**	1	0	**1**	**0**	1	1	0	**1**	1	**1**	**0**	**0**	0	...
Fred	0	1	**0**	1	0	**0**	**1**	1	1	0	**0**	1	**0**	**1**	**0**	0	...
Gábor	0	1	**1**	1	0	**1**	**1**	1	1	0	**1**	1	**0**	**0**	**1**	0	...
Henry	0	1	**0**	1	0	**1**	**1**	1	1	0	**0**	1	**0**	**1**	**1**	0	...
Copy	0	1	**1**	1	0	**1**	**0**	1	1	0	**1**	1	**0**	**1**	**0**	0	...

TU/e

Collusion-resistant traitor tracing

Schemes resistant against collusion attacks

User	Copyrighted content (fingerprinted)																
Antonino	0	1	?	1	0	?	?	1	1	0	?	1	?	?	?	0	...
Boris	0	1	?	1	0	?	?	1	1	0	?	1	?	?	?	0	...
Caroline	0	1	?	1	0	?	?	1	1	0	?	1	?	?	?	0	...
David	0	1	?	1	0	?	?	1	1	0	?	1	?	?	?	0	...
Eve	0	1	?	1	0	?	?	1	1	0	?	1	?	?	?	0	...
Fred	0	1	?	1	0	?	?	1	1	0	?	1	?	?	?	0	...
Gábor	0	1	?	1	0	?	?	1	1	0	?	1	?	?	?	0	...
Henry	0	1	?	1	0	?	?	1	1	0	?	1	?	?	?	0	...
Copy	0	1	?	1	0	?	?	1	1	0	?	1	?	?	?	0	...

TU/e

Collusion-resistant traitor tracing

Schemes resistant against collusion attacks

User	Copyrighted content (fingerprinted)																
Antonino	0	1	?	1	0	?	?	1	1	0	?	1	?	?	?	0	...
Boris	0	1	?	1	0	?	?	1	1	0	?	1	?	?	?	0	...
Caroline	0	1	?	1	0	?	?	1	1	0	?	1	?	?	?	0	...
David	0	1	?	1	0	?	?	1	1	0	?	1	?	?	?	0	...
Eve	0	1	?	1	0	?	?	1	1	0	?	1	?	?	?	0	...
Fred	0	1	?	1	0	?	?	1	1	0	?	1	?	?	?	0	...
Gábor	0	1	?	1	0	?	?	1	1	0	?	1	?	?	?	0	...
Henry	0	1	?	1	0	?	?	1	1	0	?	1	?	?	?	0	...
Copy	0	1	?	1	0	?	?	1	1	0	?	1	?	?	?	0	...

1. An algorithm to construct collusion-resistant codes

TU/e

Collusion-resistant traitor tracing

Schemes resistant against collusion attacks

User	Copyrighted content (fingerprinted)																
Antonino	0	1	?	1	0	?	?	1	1	0	?	1	?	?	?	0	...
Boris	0	1	?	1	0	?	?	1	1	0	?	1	?	?	?	0	...
Caroline	0	1	?	1	0	?	?	1	1	0	?	1	?	?	?	0	...
David	0	1	?	1	0	?	?	1	1	0	?	1	?	?	?	0	...
Eve	0	1	?	1	0	?	?	1	1	0	?	1	?	?	?	0	...
Fred	0	1	?	1	0	?	?	1	1	0	?	1	?	?	?	0	...
Gábor	0	1	?	1	0	?	?	1	1	0	?	1	?	?	?	0	...
Henry	0	1	?	1	0	?	?	1	1	0	?	1	?	?	?	0	...
Copy	0	1	?	1	0	?	?	1	1	0	?	1	?	?	?	0	...

1. An algorithm to construct collusion-resistant codes
2. An algorithm to trace pirate copies to colluders

TU/e

Collusion-resistant traitor tracing

Schemes resistant against collusion attacks

User	Copyrighted content (fingerprinted)										
Antonino	?		?	?		?		?	?	?	...
Boris	?		?	?		?		?	?	?	...
Caroline	?		?	?		?		?	?	?	...
David	?		?	?		?		?	?	?	...
Eve	?		?	?		?		?	?	?	...
Fred	?		?	?		?		?	?	?	...
Gábor	?		?	?		?		?	?	?	...
Henry	?		?	?		?		?	?	?	...
Copy	?		?	?		?		?	?	?	...

1. An algorithm to construct collusion-resistant codes
2. An algorithm to trace pirate copies to colluders

TU/e

Collusion-resistant traitor tracing

Schemes resistant against collusion attacks

User	Copyrighted content (fingerprinted)	
Antonino	$X \in \{0,1\}^{n \times \ell}$	...
Boris		...
Caroline		...
David		...
Eve		...
Fred		...
Gábor		...
Henry		...
Copy	$y \in \{0,1\}^{\ell}$	...

1. An algorithm to construct collusion-resistant codes
2. An algorithm to trace pirate copies to colluders

TU/e

Collusion-resistant traitor tracing

Schemes resistant against collusion attacks

1. An algorithm to construct collusion-resistant codes
2. An algorithm to trace pirate copies to colluders

TU/e

Collusion-resistant traitor tracing

Schemes resistant against collusion attacks

1. An algorithm to construct collusion-resistant codes

2. An algorithm to trace pirate copies to colluders

TU/e

Score-based construction

Overview

1. An algorithm to construct collusion-resistant codes

2. An algorithm to trace pirate copies to colluders

TU/e

Score-based construction

Overview

1. An algorithm to construct collusion-resistant codes
   1a. For each segment i, generate $p_i \sim F$.
   - Many values of p_i close to 0 and 1.
   - Hide choice of p_i from pirates.

2. An algorithm to trace pirate copies to colluders

TU/e

Score-based construction

Overview

1. An algorithm to construct collusion-resistant codes
   1a. For each segment i, generate $p_i \sim F$.
      - Many values of p_i close to 0 and 1.
      - Hide choice of p_i from pirates.
   1b. For each segment i, user j, choose $X_{j,i} = 1$ with prob. p_i.
2. An algorithm to trace pirate copies to colluders

TU/e

Score-based construction

Overview

1. An algorithm to construct collusion-resistant codes
   - 1a. For each segment i, generate $p_i \sim F$.
     - Many values of p_i close to 0 and 1.
     - Hide choice of p_i from pirates.
   - 1b. For each segment i, user j, choose $X_{j,i} = 1$ with prob. p_i.
2. An algorithm to trace pirate copies to colluders
   - 2a. For each segment i, user j, calculate $S_{j,i} = g(X_{j,i}, y_i, p_i)$.
     - Positive scores ($S_{j,i} > 0$) for matches ($X_{j,i} = y_i$).
     - Negative scores ($S_{j,i} < 0$) for differences ($X_{j,i} \neq y_i$).
     - Large scores ($|S_{j,i}| \gg 0$) for rare events.

TU/e

Score-based construction

Overview

1. An algorithm to construct collusion-resistant codes
   - 1a. For each segment i, generate $p_i \sim F$.
     - Many values of p_i close to 0 and 1.
     - Hide choice of p_i from pirates.
   - 1b. For each segment i, user j, choose $X_{j,i} = 1$ with prob. p_i.
2. An algorithm to trace pirate copies to colluders
   - 2a. For each segment i, user j, calculate $S_{j,i} = g(X_{j,i}, y_i, p_i)$.
     - Positive scores ($S_{j,i} > 0$) for matches ($X_{j,i} = y_i$).
     - Negative scores ($S_{j,i} < 0$) for differences ($X_{j,i} \neq y_i$).
     - Large scores ($|S_{j,i}| \gg 0$) for rare events.
   - 2b. For each user j, accuse user j iff $\sum_i S_{j,i}$ is "large".

TU/e

Score-based construction

Codewords

p_i	p_1	p_2	p_3	p_4	p_5	$\cdots$	p_{1208}
Antonino	$X_{1,1}$	$X_{1,2}$	$X_{1,3}$	$X_{1,4}$	$X_{1,5}$	$\cdots$	$X_{1,1208}$
Boris	$X_{2,1}$	$X_{2,2}$	$X_{2,3}$	$X_{2,4}$	$X_{2,5}$	$\cdots$	$X_{2,1208}$
Caroline	$X_{3,1}$	$X_{3,2}$	$X_{3,3}$	$X_{3,4}$	$X_{3,5}$	$\cdots$	$X_{3,1208}$
David	$X_{4,1}$	$X_{4,2}$	$X_{4,3}$	$X_{4,4}$	$X_{4,5}$	$\cdots$	$X_{4,1208}$
Eve	$X_{5,1}$	$X_{5,2}$	$X_{5,3}$	$X_{5,4}$	$X_{5,5}$	$\cdots$	$X_{5,1208}$
Fred	$X_{6,1}$	$X_{6,2}$	$X_{6,3}$	$X_{6,4}$	$X_{6,5}$	$\cdots$	$X_{6,1208}$
Gábor	$X_{7,1}$	$X_{7,2}$	$X_{7,3}$	$X_{7,4}$	$X_{7,5}$	$\cdots$	$X_{7,1208}$
Henry	$X_{8,1}$	$X_{8,2}$	$X_{8,3}$	$X_{8,4}$	$X_{8,5}$	$\cdots$	$X_{8,1208}$
Copy	y_1	y_2	y_3	y_4	y_5	$\cdots$	y_{1208}

TU/e

Score-based construction

Codewords

1a. For each segment i, generate $p_i \sim F$.

p_i	0.20	0.05	0.88	0.79	0.98	...	0.18
Antonino	$X_{1,1}$	$X_{1,2}$	$X_{1,3}$	$X_{1,4}$	$X_{1,5}$	...	$X_{1,1208}$
Boris	$X_{2,1}$	$X_{2,2}$	$X_{2,3}$	$X_{2,4}$	$X_{2,5}$	...	$X_{2,1208}$
Caroline	$X_{3,1}$	$X_{3,2}$	$X_{3,3}$	$X_{3,4}$	$X_{3,5}$	...	$X_{3,1208}$
David	$X_{4,1}$	$X_{4,2}$	$X_{4,3}$	$X_{4,4}$	$X_{4,5}$	...	$X_{4,1208}$
Eve	$X_{5,1}$	$X_{5,2}$	$X_{5,3}$	$X_{5,4}$	$X_{5,5}$	...	$X_{5,1208}$
Fred	$X_{6,1}$	$X_{6,2}$	$X_{6,3}$	$X_{6,4}$	$X_{6,5}$	...	$X_{6,1208}$
Gábor	$X_{7,1}$	$X_{7,2}$	$X_{7,3}$	$X_{7,4}$	$X_{7,5}$	...	$X_{7,1208}$
Henry	$X_{8,1}$	$X_{8,2}$	$X_{8,3}$	$X_{8,4}$	$X_{8,5}$	...	$X_{8,1208}$
Copy	y_1	y_2	y_3	y_4	y_5	...	y_{1208}

TU/e

Score-based construction

Codewords

1b. For each segment i, user j, choose $X_{j,i} = 1$ with prob. p_i.

p_i	0.20	0.05	0.88	0.79	0.98	...	0.18
Antonino	0	0	1	1	1	...	0
Boris	1	0	1	1	1	...	1
Caroline	1	0	0	1	0	...	0
David	0	0	1	1	1	...	0
Eve	0	0	1	0	1	...	0
Fred	1	0	1	0	1	...	0
Gábor	0	0	1	0	1	...	0
Henry	1	0	0	0	1	...	0
Copy	y_1	y_2	y_3	y_4	y_5	...	y_{1208}

TU/e

Score-based construction

Coalition

Pirates get their versions, ...

p_i	.	.	.	.	.	...	.
Antonino	.	.	.	.	.	...	.
Boris	.	.	.	.	.	...	.
Caroline	1	0	0	1	0	...	0
David	.	.	.	.	.	...	.
Eve	0	0	1	0	1	...	0
Fred	.	.	.	.	.	...	.
Gábor	.	.	.	.	.	...	.
Henry	1	0	0	0	1	...	0
Copy	y_1	y_2	y_3	y_4	y_5	...	y_{1208}

$$\text{Coalition} = \{\text{Caroline}, \text{Eve}, \text{Henry}\}$$

TU/e

Score-based construction

Coalition

Pirates get their versions, compare them ...

p_i	.	.	.	.	.	...	.
Antonino	.	.	.	.	.	...	.
Boris	.	.	.	.	.	...	.
Caroline	**1**	0	**0**	**1**	**0**	...	0
David	.	.	.	.	.	...	.
Eve	**0**	0	**1**	**0**	**1**	...	0
Fred	.	.	.	.	.	...	.
Gábor	.	.	.	.	.	...	.
Henry	**1**	0	**0**	**0**	**1**	...	0
Copy	y_1	y_2	y_3	y_4	y_5	...	y_{1208}

$$\text{Coalition} = \{\text{Caroline}, \text{Eve}, \text{Henry}\}$$

TU/e

Score-based construction

Coalition

Pirates get their versions, compare them and make a copy.

p_i	.	.	.	.	.	...	.
Antonino	.	.	.	.	.	...	.
Boris	.	.	.	.	.	...	.
Caroline	**1**	0	**0**	**1**	**0**	...	0
David	.	.	.	.	.	...	.
Eve	**0**	0	**1**	**0**	**1**	...	0
Fred	.	.	.	.	.	...	.
Gábor	.	.	.	.	.	...	.
Henry	**1**	0	**0**	**0**	**1**	...	0
Copy	**0**	0	**0**	**1**	**1**	...	0

Coalition $= \{\text{Caroline}, \text{Eve}, \text{Henry}\}$

TU/e

Score-based construction

Scores

The copy is distributed and detected by the tracer.

p_i	0.20	0.05	0.88	0.79	0.98	...	0.18
Antonino	0	0	1	1	1	...	0
Boris	1	0	1	1	1	...	1
Caroline	1	0	0	1	0	...	0
David	0	0	1	1	1	...	0
Eve	0	0	1	0	1	...	0
Fred	1	0	1	0	1	...	0
Gábor	0	0	1	0	1	...	0
Henry	1	0	0	0	1	...	0
Copy	0	0	0	1	1	...	0

$$\text{Coalition} = \{\text{Caroline}, \text{Eve}, \text{Henry}\}$$

TU/e

Score-based construction

Scores

2a. For each segment i, user j, calculate $S_{j,i} = g(X_{j,i}, y_i, p_i)$.

p_i	0.20	0.05	0.88	0.79	0.98	...	0.18
Antonino	+0.5	+0.2	−0.4	+0.5	+0.1	...	+0.5
Boris	−2.0	+0.2	−0.4	+0.5	+0.1	...	−2.1
Caroline	−2.0	+0.2	+2.7	+0.5	−7.2	...	+0.5
David	+0.5	+0.2	−0.4	+0.5	+0.1	...	+0.5
Eve	+0.5	+0.2	−0.4	−1.9	+0.1	...	+0.5
Fred	−2.0	+0.2	−0.4	−1.9	+0.1	...	+0.5
Gábor	+0.5	+0.2	−0.4	−1.9	+0.1	...	+0.5
Henry	−2.0	+0.2	+2.7	−1.9	+0.1	...	+0.5
Copy	0	0	0	1	1	...	0

$$\text{Coalition} = \{\text{Caroline}, \text{Eve}, \text{Henry}\}$$

TU/e

Score-based construction

Scores

2b. For each user j, accuse user j iff $\sum_i S_{j,i}$ is "large".

p_i	0.20	0.05	0.88	0.79	0.98	...	0.18	$\sum_i S_{j,i}$
Antonino	+0.5	+0.2	−0.4	+0.5	+0.1	...	+0.5	0
Boris	−2.0	+0.2	−0.4	+0.5	+0.1	...	−2.1	0
Caroline	−2.0	+0.2	+2.7	+0.5	−7.2	...	+0.5	0
David	+0.5	+0.2	−0.4	+0.5	+0.1	...	+0.5	0
Eve	+0.5	+0.2	−0.4	−1.9	+0.1	...	+0.5	0
Fred	−2.0	+0.2	−0.4	−1.9	+0.1	...	+0.5	0
Gábor	+0.5	+0.2	−0.4	−1.9	+0.1	...	+0.5	0
Henry	−2.0	+0.2	+2.7	−1.9	+0.1	...	+0.5	0
Copy	0	0	0	1	1	...	0	

$$\text{Coalition} = \{\text{Caroline}, \text{Eve}, \text{Henry}\}$$

TU/e

Score-based construction

Scores

2b. For each user j, accuse user j iff $\sum_i S_{j,i}$ is "large".

p_i	0.20	0.05	0.88	0.79	0.98	...	0.18	$\sum_i S_{j,i}$
Antonino	+0.5	+0.2	−0.4	+0.5	+0.1	...	+0.5	+14
Boris	−2.0	+0.2	−0.4	+0.5	+0.1	...	−2.1	−19
Caroline	−2.0	+0.2	+2.7	+0.5	−7.2	...	+0.5	+291
David	+0.5	+0.2	−0.4	+0.5	+0.1	...	+0.5	+29
Eve	+0.5	+0.2	−0.4	−1.9	+0.1	...	+0.5	+292
Fred	−2.0	+0.2	−0.4	−1.9	+0.1	...	+0.5	−53
Gábor	+0.5	+0.2	−0.4	−1.9	+0.1	...	+0.5	−42
Henry	−2.0	+0.2	+2.7	−1.9	+0.1	...	+0.5	+269
Copy	0	0	0	1	1	...	0	

$$\text{Coalition} = \{\text{Caroline}, \text{Eve}, \text{Henry}\}$$

TU/e

Score-based construction

Scores

2b. For each user j, accuse user j iff $\sum_i S_{j,i}$ is "large".

p_i	0.20	0.05	0.88	0.79	0.98	...	0.18	$\sum_i S_{j,i}$
Antonino	+0.5	+0.2	−0.4	+0.5	+0.1	...	+0.5	+14
Boris	−2.0	+0.2	−0.4	+0.5	+0.1	...	−2.1	−19
Caroline	−2.0	+0.2	+2.7	+0.5	−7.2	...	+0.5	+291
David	+0.5	+0.2	−0.4	+0.5	+0.1	...	+0.5	+29
Eve	+0.5	+0.2	−0.4	−1.9	+0.1	...	+0.5	+292
Fred	−2.0	+0.2	−0.4	−1.9	+0.1	...	+0.5	−53
Gábor	+0.5	+0.2	−0.4	−1.9	+0.1	...	+0.5	−42
Henry	−2.0	+0.2	+2.7	−1.9	+0.1	...	+0.5	+269
Copy	0	0	0	1	1	...	0	

$$\text{Coalition} = \{\text{Caroline, Eve, Henry}\}$$

$$\text{Accused} = \{\text{Caroline, Eve, Henry}\}$$

TU/e

Score-based construction

Overview

1. An algorithm to construct collusion-resistant codes
   - 1a. For each segment i, generate $p_i \sim F$.
     - Many values of p_i close to 0 and 1.
     - Hide choice of p_i from pirates.
   - 1b. For each segment i, user j, choose $X_{j,i} = 1$ with prob. p_i.
2. An algorithm to trace pirate copies to colluders
   - 2a. For each segment i, user j, calculate $S_{j,i} = g(X_{j,i}, y_i, p_i)$.
     - Positive scores ($S_{j,i} > 0$) for matches ($X_{j,i} = y_i$).
     - Negative scores ($S_{j,i} < 0$) for differences ($X_{j,i} \neq y_i$).
     - Large scores ($|S_{j,i}| \gg 0$) for rare events.
   - 2b. For each user j, accuse user j iff $\sum_i S_{j,i}$ is "large".

TU/e

Score-based construction

Overview

1. An algorithm to construct collusion-resistant codes
   - 1a. For each segment i, generate $p_i \sim F$.
     - Many values of p_i close to 0 and 1.
     - Hide choice of p_i from pirates.
   - 1b. For each segment i, user j, choose $X_{j,i} = 1$ with prob. p_i.
2. An algorithm to trace pirate copies to colluders
   - 2a. For each segment i, user j, calculate $S_{j,i} = g(X_{j,i}, y_i, p_i)$.
     - Positive scores ($S_{j,i} > 0$) for matches ($X_{j,i} = y_i$).
     - Negative scores ($S_{j,i} < 0$) for differences ($X_{j,i} \neq y_i$).
     - Large scores ($|S_{j,i}| \gg 0$) for rare events.
   - 2b. For each user j, accuse user j iff $\sum_i S_{j,i}$ is "large".

What does the code length become when we optimize F and g?

TU/e

Fighting against specific attacks

Arbitrary attacks

1. An algorithm to construct collusion-resistant codes
   - 1a. For each segment i, generate $p_i \sim F(p_i) = \frac{2}{\pi} \arcsin \sqrt{p_i}$.
   - 1b. For each segment i, user j, choose $X_{j,i} = 1$ with prob. p_i.
2. An algorithm to trace pirate copies to colluders
   - 2a. For each segment i, user j, calculate $S_{j,i} = g(X_{j,i}, y_i, p_i)$.

$$g(X_{j,i}, y_i, p_i) = \begin{cases} +p/(1-p) & (X_{j,i}, y_i) = (0,0) \\ -1 & (X_{j,i}, y_i) = (0,1) \\ -1 & (X_{j,i}, y_i) = (1,0) \\ +(1-p)/p & (X_{j,i}, y_i) = (1,1) \end{cases}$$

   - 2b. For each user j, accuse user j iff $\sum_i S_{j,i}$ is "large".

$$\ell \sim 2c^2 \ln n$$

TU/e

Fighting against specific attacks

The interleaving attack

1. An algorithm to construct collusion-resistant codes
   - 1a. For each segment i, generate $p_i \sim F(p_i) = \frac{2}{\pi} \arcsin \sqrt{p_i}$.
   - 1b. For each segment i, user j, choose $X_{j,i} = 1$ with prob. p_i.
2. An algorithm to trace pirate copies to colluders
   - 2a. For each segment i, user j, calculate $S_{j,i} = g(X_{j,i}, y_i, p_i)$.

$$g(X_{j,i}, y_i, p_i) = \begin{cases} +p/(1-p) & (X_{j,i}, y_i) = (0,0) \\ -1 & (X_{j,i}, y_i) = (0,1) \\ -1 & (X_{j,i}, y_i) = (1,0) \\ +(1-p)/p & (X_{j,i}, y_i) = (1,1) \end{cases}$$

   - 2b. For each user j, accuse user j iff $\sum_i S_{j,i}$ is "large".

$$\ell \sim 2c^2 \ln n$$

TU/e

Fighting against specific attacks

The all-1 attack

1. An algorithm to construct collusion-resistant codes
   - 1a. For each segment i, generate $p_i \sim F(p_i) = \frac{2}{\pi} \arcsin \sqrt{p_i}$.
   - 1b. For each segment i, user j, choose $X_{j,i} = 1$ with prob. p_i.
2. An algorithm to trace pirate copies to colluders
   - 2a. For each segment i, user j, calculate $S_{j,i} = g(X_{j,i}, y_i, p_i)$.

$$g(X_{j,i}, y_i, p_i) = \begin{cases} +p/(1-p) & (X_{j,i}, y_i) = (0,0) \\ -p(1-p)^{c-1}/(1-(1-p)^c) & (X_{j,i}, y_i) = (0,1) \\ -1 & (X_{j,i}, y_i) = (1,0) \\ +(1-p)^c/(1-(1-p)^c) & (X_{j,i}, y_i) = (1,1) \end{cases}$$

   - 2b. For each user j, accuse user j iff $\sum_i S_{j,i}$ is "large".

$$\ell = O(c^{1.5} \ln n)$$

TU/e

Fighting against specific attacks

The all-1 attack

1. An algorithm to construct collusion-resistant codes
   1a. For each segment i, generate $p_i \equiv p = O(1/c)$.
   1b. For each segment i, user j, choose $X_{j,i} = 1$ with prob. p_i.
2. An algorithm to trace pirate copies to colluders
   2a. For each segment i, user j, calculate $S_{j,i} = g(X_{j,i}, y_i, p_i)$.

$$g(X_{j,i}, y_i, p_i) = \begin{cases} +p/(1-p) & (X_{j,i}, y_i) = (0,0) \\ -p(1-p)^{c-1}/(1-(1-p)^c) & (X_{j,i}, y_i) = (0,1) \\ -\infty & (X_{j,i}, y_i) = (1,0) \\ +(1-p)^c/(1-(1-p)^c) & (X_{j,i}, y_i) = (1,1) \end{cases}$$

   2b. For each user j, accuse user j iff $\sum_i S_{j,i}$ is "large".

$$\ell \sim 2c \ln n$$

TU/e

Fighting against specific attacks

The minority voting attack

1. An algorithm to construct collusion-resistant codes
   - 1a. For each segment i, generate $p_i \sim F(p_i) = \frac{2}{\pi} \arcsin \sqrt{p_i}$.
   - 1b. For each segment i, user j, choose $X_{j,i} = 1$ with prob. p_i.
2. An algorithm to trace pirate copies to colluders
   - 2a. For each segment i, user j, calculate $S_{j,i} = g(X_{j,i}, y_i, p_i)$.

$$g(X_{j,i}, y_i, p_i) \approx \begin{cases} +\ldots & (X_{j,i}, y_i) = (0,0) \\ -\ldots & (X_{j,i}, y_i) = (0,1) \\ -\ldots & (X_{j,i}, y_i) = (1,0) \\ +\ldots & (X_{j,i}, y_i) = (1,1) \end{cases}$$

   - 2b. For each user j, accuse user j iff $\sum_i S_{j,i}$ is "large".

$$\ell = O(c^{1.5} \ln n)$$

TU/e

Fighting against specific attacks

The minority voting attack

1. An algorithm to construct collusion-resistant codes
   - 1a. For each segment i, generate $p_i \equiv p = O(1/c)$.
   - 1b. For each segment i, user j, choose $X_{j,i} = 1$ with prob. p_i.
2. An algorithm to trace pirate copies to colluders
   - 2a. For each segment i, user j, calculate $S_{j,i} = g(X_{j,i}, y_i, p_i)$.

$$g(X_{j,i}, y_i, p_i) \approx \begin{cases} +p/(1-p) & (X_{j,i}, y_i) = (0,0) \\ -p(1-p)^{c-1}/(1-(1-p)^c) & (X_{j,i}, y_i) = (0,1) \\ -1 & (X_{j,i}, y_i) = (1,0) \\ +(1-p)^c/(1-(1-p)^c) & (X_{j,i}, y_i) = (1,1) \end{cases}$$

   - 2b. For each user j, accuse user j iff $\sum_i S_{j,i}$ is "large".

$$\ell \sim 2c \ln n$$

TU/e

Fighting against specific attacks

The majority voting attack

1. An algorithm to construct collusion-resistant codes
   - 1a. For each segment i, generate $p_i \sim F(p_i) = \frac{2}{\pi} \arcsin \sqrt{p_i}$.
   - 1b. For each segment i, user j, choose $X_{j,i} = 1$ with prob. p_i.
2. An algorithm to trace pirate copies to colluders
   - 2a. For each segment i, user j, calculate $S_{j,i} = g(X_{j,i}, y_i, p_i)$.

$$g(X_{j,i}, y_i, p_i) = \begin{cases} +\ldots & (X_{j,i}, y_i) = (0,0) \\ -\ldots & (X_{j,i}, y_i) = (0,1) \\ -\ldots & (X_{j,i}, y_i) = (1,0) \\ +\ldots & (X_{j,i}, y_i) = (1,1) \end{cases}$$

   - 2b. For each user j, accuse user j iff $\sum_i S_{j,i}$ is "large".

$$\ell = O(c^{1.5} \ln n)$$

TU/e

Fighting against specific attacks

The majority voting attack

1. An algorithm to construct collusion-resistant codes
   - 1a. For each segment i, generate $p_i \equiv p = 1/2$.
   - 1b. For each segment i, user j, choose $X_{j,i} = 1$ with prob. p_i.
2. An algorithm to trace pirate copies to colluders
   - 2a. For each segment i, user j, calculate $S_{j,i} = g(X_{j,i}, y_i, p_i)$.

$$g(X_{j,i}, y_i, p_i) = \begin{cases} +1 & (X_{j,i}, y_i) = (0,0) \\ -1 & (X_{j,i}, y_i) = (0,1) \\ -1 & (X_{j,i}, y_i) = (1,0) \\ +1 & (X_{j,i}, y_i) = (1,1) \end{cases}$$

   - 2b. For each user j, accuse user j iff $\sum_i S_{j,i}$ is "large".

$$\ell \sim \pi c \ln n$$

TU/e

Fighting against specific attacks

The coin-flip attack

1. An algorithm to construct collusion-resistant codes
   - 1a. For each segment i, generate $p_i \sim F(p_i) = \frac{2}{\pi} \arcsin \sqrt{p_i}$.
   - 1b. For each segment i, user j, choose $X_{j,i} = 1$ with prob. p_i.
2. An algorithm to trace pirate copies to colluders
   - 2a. For each segment i, user j, calculate $S_{j,i} = g(X_{j,i}, y_i, p_i)$.

$$g(X_{j,i}, y_i, p_i) = \left(p^{c-1} + (1-p)^{c-1}\right) \times \begin{cases} +p/(1-p^c+(1-p)^c) & (X_{j,i}, y_i) = (0,0) \\ -p/(1+p^c-(1-p)^c) & (X_{j,i}, y_i) = (0,1) \\ -(1-p)/(1-p^c+(1-p)^c) & (X_{j,i}, y_i) = (1,0) \\ +(1-p)/(1+p^c-(1-p)^c) & (X_{j,i}, y_i) = (1,1) \end{cases}$$

   - 2b. For each user j, accuse user j iff $\sum_i S_{j,i}$ is "large".

$$\ell = O(c^{1.5} \ln n)$$

TU/e

Fighting against specific attacks

The coin-flip attack

1. An algorithm to construct collusion-resistant codes
   - 1a. For each segment i, generate $p_i \equiv p = O(1/c)$.
   - 1b. For each segment i, user j, choose $X_{j,i} = 1$ with prob. p_i.
2. An algorithm to trace pirate copies to colluders
   - 2a. For each segment i, user j, calculate $S_{j,i} = g(X_{j,i}, y_i, p_i)$.

$$g(X_{j,i}, y_i, p_i) = \left(p^{c-1} + (1-p)^{c-1}\right) \times \begin{cases} +p/(1-p^c+(1-p)^c) & (X_{j,i}, y_i) = (0,0) \\ -p/(1+p^c-(1-p)^c) & (X_{j,i}, y_i) = (0,1) \\ -(1-p)/(1-p^c+(1-p)^c) & (X_{j,i}, y_i) = (1,0) \\ +(1-p)/(1+p^c-(1-p)^c) & (X_{j,i}, y_i) = (1,1) \end{cases}$$

   - 2b. For each user j, accuse user j iff $\sum_i S_{j,i}$ is "large".

$$\ell \sim 4c \ln n$$

TU/e

Results

The Tardos scheme

Table: Asymptotics of $L = \dfrac{\ell}{\ln n}$ for large n.

	Efficient constr.		Lower bounds	
Arbitrary attacks	$100c^2$	[Tar'03]	$\Omega(c^2)$	[Tar'03]
Interleaving attack	$100c^2$	[Tar'03]	$\Omega(c)$	
All-1 attack	$100c^2$	[Tar'03]	$\Omega(c)$	
Minority voting	$100c^2$	[Tar'03]	$\Omega(c)$	
Majority voting	$100c^2$	[Tar'03]	$\Omega(c)$	
Coin-flip attack	$100c^2$	[Tar'03]	$\Omega(c)$	

TU/e

Results

Improvements of the Tardos scheme

Table: Asymptotics of $L = \dfrac{\ell}{\ln n}$ for large n.

	Efficient constr.		Lower bounds	
Arbitrary attacks	$2c^2$	[ODS'13]	$2c^2$	[HM'12]
Interleaving attack	$2c^2$	[ODS'13]	$2c^2$	[HM'12]
All-1 attack	$O(c^{1.5})$	[ODS'13]	$\Omega(c)$	
Minority voting	$O(c^{1.5})$	[ODS'13]	$\Omega(c)$	
Majority voting	$O(c^{1.5})$	[ODS'13]	$\Omega(c)$	
Coin-flip attack	$O(c^{1.5})$	[ODS'13]	$\Omega(c)$	

TU/e

Results

Results from group testing

Table: Asymptotics of $L = \dfrac{\ell}{\ln n}$ for large n.

	Efficient constr.		Lower bounds	
Arbitrary attacks	$2c^2$	[ODS'13]	$2c^2$	[HM'12]
Interleaving attack	$2c^2$	[ODS'13]	$2c^2$	[HM'12]
All-1 attack	ec	[C+'11]	$\log_2(e)c$	[Seb'85]
Minority voting	$O(c^{1.5})$	[ODS'13]	$\Omega(c)$	
Majority voting	$O(c^{1.5})$	[ODS'13]	$\Omega(c)$	
Coin-flip attack	$O(c^{1.5})$	[ODS'13]	$\Omega(c)$	

TU/e

Results

Contributions

Table: Asymptotics of $L = \dfrac{\ell}{\ln n}$ for large n.

	Efficient constr.		Lower bounds	
Arbitrary attacks	$2c^2$	[ODS'13]	$2c^2$	[HM'12]
Interleaving attack	$2c^2$	[ODS'13]	$2c^2$	[HM'12]
All-1 attack	$2c$	[Laa'13]	$\log_2(e)c$	[Seb'85]
Minority voting	$2c$	[Laa'13]	$\Omega(c)$	
Majority voting	πc	[Laa'13]	$\Omega(c)$	
Coin-flip attack	$4c$	[Laa'13]	$\Omega(c)$	

TU/e

Conclusion

If you do know the pirate strategy...

- ...you can find pirates much faster!
- Trick: Optimize g, then optimize and fix p
- Code length often linear in c, decreases linearly in q
- Applications to group testing

If you don't know the pirate strategy...

- ...use the interleaving defense, also dynamically!
- Statically optimal, dynamically possibly optimal(?)
- Seems to work well in practice (simulations)

TU/e

Questions?