Progress in Traitor Tracing with Applications to Group Testing

Thijs Laarhoven

(Joint work with Jeroen Doumen, Jan-Jaap Oosterwijk, Peter Roelse, Boris Škorić, Benne de Weger)

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

Champaign, Illinois, USA
(October 1, 2013)

Outline

Problem: 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	...

Problem: 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	...

Solution: Embed 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	...

Solution: Embed 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

Solution: Embed 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	...

Solution: Embed 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	...

Problem: 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

Problem: 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

Problem: 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 ...

Solution: Collusion-resistant schemes

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	...

Solution: Collusion-resistant schemes

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

Solution: Collusion-resistant schemes

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

Solution: Collusion-resistant schemes

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

Solution: Collusion-resistant schemes

User	Copyrighted content (fingerprinted)	
Antonino		...
Boris		...
Caroline	$X \in \{0,1\}^{n \times \ell}$	...
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

Solution: Collusion-resistant schemes

User	Copyrighted content (fingerprinted)	
Antonino		...
Boris		...
Caroline	$X \in \{0,1\}^{n \times \ell}$	...
David		...
Eve		...
Fred	$\left(\text{[Tar03]: } \ell = L \cdot c^2 \log n/\varepsilon\right)$	...
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

Solution: Collusion-resistant schemes

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

Solution: Collusion-resistant schemes

1. An algorithm to construct collusion-resistant codes

2. An algorithm to trace pirate copies to colluders

The Tardos scheme: Overview

1. An algorithm to construct collusion-resistant codes

2. An algorithm to trace pirate copies to colluders

The Tardos scheme: 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

The Tardos scheme: 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

The Tardos scheme: 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.

The Tardos scheme: 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".

The Tardos scheme: Codewords

p_i	p_1	p_2	p_3	p_4	p_5	$\cdots$	p_{1200}
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}

The Tardos scheme: Codewords

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

p_i	p_1	p_2	p_3	p_4	p_5	$\cdots$	p_{1200}
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

The Tardos scheme: 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}

The Tardos scheme: 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	$X_{1,1}$	$X_{1,2}$	$X_{1,3}$	$X_{1,4}$	$X_{1,5}$	...	$X_{1,1208}$
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

The Tardos scheme: 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}

The Tardos scheme: 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}

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

The Tardos scheme: 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}

Coalition $=$ {Caroline, Eve, Henry}

The Tardos scheme: 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 $=$ {Caroline, Eve, Henry}

The Tardos scheme: 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

Coalition = {Caroline, Eve, Henry}

The Tardos scheme: 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	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

Coalition = {Caroline, Eve, Henry}

The Tardos scheme: 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

Coalition = {Caroline, Eve, Henry}

TU/e

The Tardos scheme: 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	

Coalition = {Caroline, Eve, Henry}

The Tardos scheme: 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	

Coalition $=$ {Caroline, Eve, Henry}

TU/e

The Tardos scheme: 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	

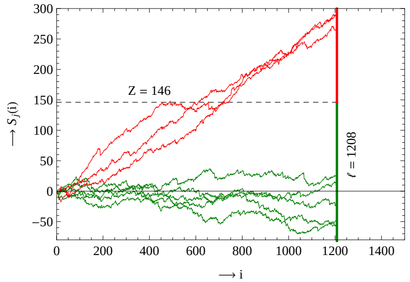
$$\text{Coalition} = \{\text{Caroline, Eve, Henry}\}$$

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

The Tardos scheme: 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								$+14$
Boris								-19
Caroline								$+291$
David								$+29$
Eve								$+292$
Fred								-53
Gábor								-42
Henry								$+269$
Copy	0	0	0	1	1	...	0	



Coalition = {Caroline, Eve, Henry}

Accused = {Caroline, Eve, Henry}

The Tardos scheme: 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".

The Tardos scheme: 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 is the best choice for F and g?

The Tardos scheme: Improvements

Table: Constructive upper bounds on L (large c asymptotics).

	[Tar03]	[S+06]	[N+07]	[BT08]	[S+08]	[N+09]	[LdW12]	[LdW13]	[O+13]
L :	100	39.5	20.6	19.7	9.9	5.4	4.9	4.9*	2
g :	g^{Tar}	g^{Tar}	g^{Tar}	g^{Tar}	g^{sym}	g^{sym}	g^{sym}	g^{sym}	g^{int}
F :	F^{arc}	F^{arc}	F^{*}	F^{arc}	F^{arc}	F^{*}	F^{arc}	F^{*}	F^{arc}

The Tardos scheme: Improvements

Table: Constructive upper bounds on L (large c asymptotics).

	[Tar03]	[S+06]	[N+07]	[BT08]	[S+08]	[N+09]	[LdW12]	[LdW13]	[O+13]
L :	100	39.5	20.6	19.7	9.9	5.4	4.9	4.9*	2
g :	g^{Tar}	g^{Tar}	g^{Tar}	g^{Tar}	g^{sym}	g^{sym}	g^{sym}	g^{sym}	g^{int}
F :	F^{arc}	F^{arc}	F^{*}	F^{arc}	F^{arc}	F^{*}	F^{arc}	F^{*}	F^{arc}

$$g^{\mathrm{Tar}}(X_{j,i}, y_i, p_i) = \begin{cases} 0, & \text{if } X_{j,i} = 0, y_i = 0, \\ -\sqrt{p_i/(1-p_i)}, & \text{if } X_{j,i} = 0, y_i = 1, \\ 0, & \text{if } X_{j,i} = 1, y_i = 0, \\ +\sqrt{(1-p_i)/p_i}, & \text{if } X_{j,i} = 1, y_i = 1. \end{cases}$$

The Tardos scheme: Improvements

Table: Constructive upper bounds on L (large c asymptotics).

	[Tar03]	[S+06]	[N+07]	[BT08]	[S+08]	[N+09]	[LdW12]	[LdW13]	[O+13]
L :	100	39.5	20.6	19.7	9.9	5.4	4.9	4.9*	2
g :	g^{Tar}	g^{Tar}	g^{Tar}	g^{Tar}	g^{sym}	g^{sym}	g^{sym}	g^{sym}	g^{int}
F :	F^{arc}	F^{arc}	F^{*}	F^{arc}	F^{arc}	F^{*}	F^{arc}	F^{*}	F^{arc}

$$g^{\mathrm{sym}}(X_{j,i}, y_i, p_i) = \begin{cases} +\sqrt{p_i/(1-p_i)}, & \text{if } X_{j,i} = 0, y_i = 0, \\ -\sqrt{p_i/(1-p_i)}, & \text{if } X_{j,i} = 0, y_i = 1, \\ -\sqrt{(1-p_i)/p_i}, & \text{if } X_{j,i} = 1, y_i = 0, \\ +\sqrt{(1-p_i)/p_i}, & \text{if } X_{j,i} = 1, y_i = 1. \end{cases}$$

TU/e

The Tardos scheme: Improvements

Table: Constructive upper bounds on L (large c asymptotics).

	[Tar03]	[S+06]	[N+07]	[BT08]	[S+08]	[N+09]	[LdW12]	[LdW13]	[O+13]
L :	100	39.5	20.6	19.7	9.9	5.4	4.9	4.9*	2
g :	g^{Tar}	g^{Tar}	g^{Tar}	g^{Tar}	g^{sym}	g^{sym}	g^{sym}	g^{sym}	g^{int}
F :	F^{arc}	F^{arc}	F^{*}	F^{arc}	F^{arc}	F^{*}	F^{arc}	F^{*}	F^{arc}

$$g^{\mathrm{int}}(X_{j,i}, y_i, p_i) = \begin{cases} +p_i/(1-p_i), & \text{if } X_{j,i} = 0, y_i = 0, \\ -1, & \text{if } X_{j,i} = 0, y_i = 1, \\ -1, & \text{if } X_{j,i} = 1, y_i = 0, \\ +(1-p_i)/p_i, & \text{if } X_{j,i} = 1, y_i = 1. \end{cases}$$

The Tardos scheme: Comparison

Table: Constructive upper bounds on L (large c asymptotics).

	[Tar03]	[S+06]	[N+07]	[BT08]	[S+08]	[N+09]	[LdW12]	[LdW13]	[O+13]
L :	100	39.5	20.6	19.7	9.9	5.4	4.9	4.9^*	2
g :	g^{Tar}	g^{Tar}	g^{Tar}	g^{Tar}	g^{sym}	g^{sym}	g^{sym}	g^{sym}	g^{int}
F :	F^{arc}	F^{arc}	F^*	F^{arc}	F^{arc}	F^*	F^{arc}	F^*	F^{arc}

Table: Information-theoretic lower bounds on L (large c asymptotics).

	[Tar03]	[AT09]	[HM09]	[HM12]
L :	$\Omega(1)$	"2"	2?	2
F :	?	?	F^{arc}?	F^{arc}
ρ :	?	?	int?	int

The Tardos scheme: Comparison

Table: Constructive upper bounds on L (large c asymptotics).

	[Tar03]	[S+06]	[N+07]	[BT08]	[S+08]	[N+09]	[LdW12]	[LdW13]	[O+13]
L :	100	39.5	20.6	19.7	9.9	5.4	4.9	4.9^*	**2**
g :	g^{Tar}	g^{Tar}	g^{Tar}	g^{Tar}	g^{sym}	g^{sym}	g^{sym}	g^{sym}	g^{int}
F :	F^{arc}	F^{arc}	F^*	F^{arc}	F^{arc}	F^*	F^{arc}	F^*	F^{arc}

Table: Information-theoretic lower bounds on L (large c asymptotics).

	[Tar03]	[AT09]	[HM09]	[HM12]
L :	$\Omega(1)$	"2"	2?	**2**
F :	?	?	F^{arc}?	F^{arc}
ρ :	?	?	int?	int

Efficient capacity-achieving construction

Main result: Using the score function $g = g^{\mathrm{int}}$ given by

$$g^{\mathrm{int}}(X_{j,i}, y_i, p_i) = \begin{cases} +p_i/(1-p_i), & \text{if } X_{j,i} = 0, y_i = 0, \\ -1, & \text{if } X_{j,i} = 0, y_i = 1, \\ -1, & \text{if } X_{j,i} = 1, y_i = 0, \\ +(1-p_i)/p_i, & \text{if } X_{j,i} = 1, y_i = 1. \end{cases}$$

and the distribution function $F = F^{\mathrm{arc}}$ given by

$$F^{\mathrm{arc}}(p_i) = \frac{2}{\pi} \arcsin \sqrt{p_i}$$

we get an efficient construction that achieves capacity for large c.

Static vs. Dynamic

Previous slides dealt with the *static setting*:

- First generate X, then y, then trace.
- Static content: video on demand, software, ...
- Optimized Tardos scheme achieves capacity.

In some practical scenarios, we have a *dynamic setting*:

- $X_1, y_1, X_2, y_2, \ldots, X_\ell, y_\ell$.
- Dynamic content: live streams of baseball, football, ...
- More efficient schemes?
- What is the capacity?

Static vs. Dynamic

Previous slides dealt with the *static setting*:

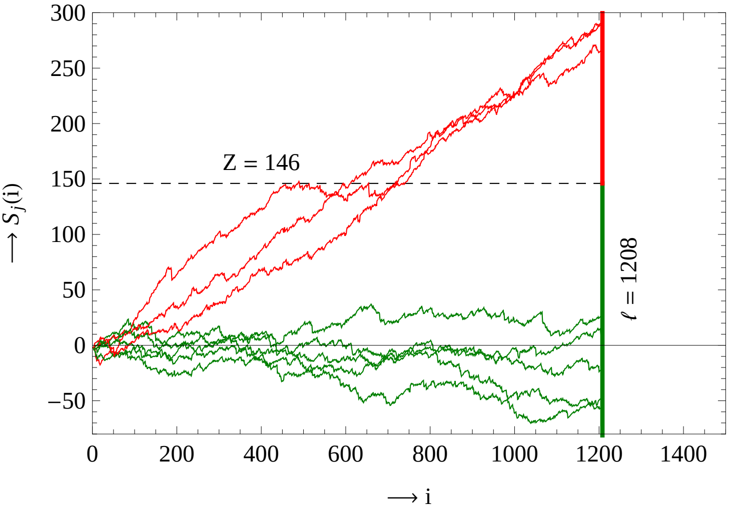
- First generate X, then y, then trace.
- Static content: video on demand, software, ...
- Optimized Tardos scheme achieves capacity.

In some practical scenarios, we have a *dynamic setting*:

- $X_1, y_1, X_2, y_2, \ldots, X_\ell, y_\ell$.
- Dynamic content: live streams of baseball, football, ...
- More efficient schemes?
- What is the capacity?

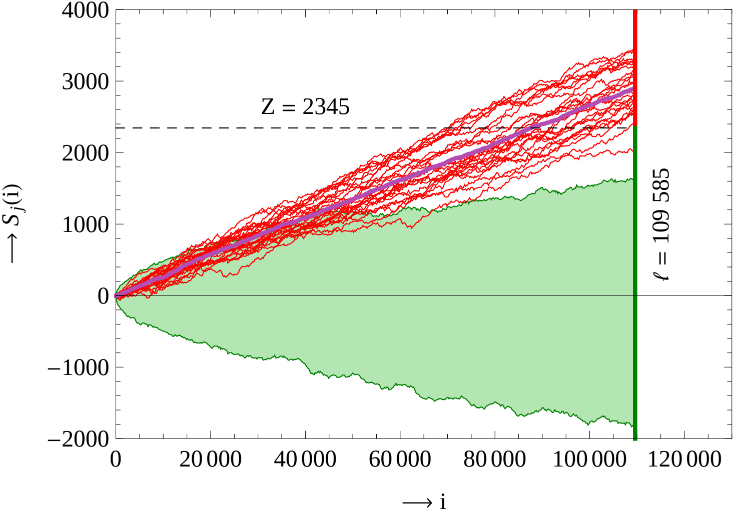
The static Tardos scheme

Earlier example, with $n = 8$ users and $c = 3$ colluders.



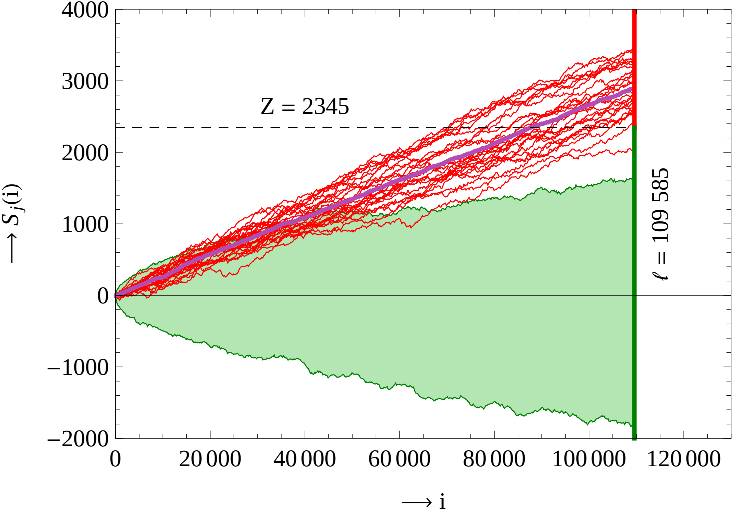
The static Tardos scheme

Bigger example, with $n = 10^6$ users and $c = 25$ colluders.



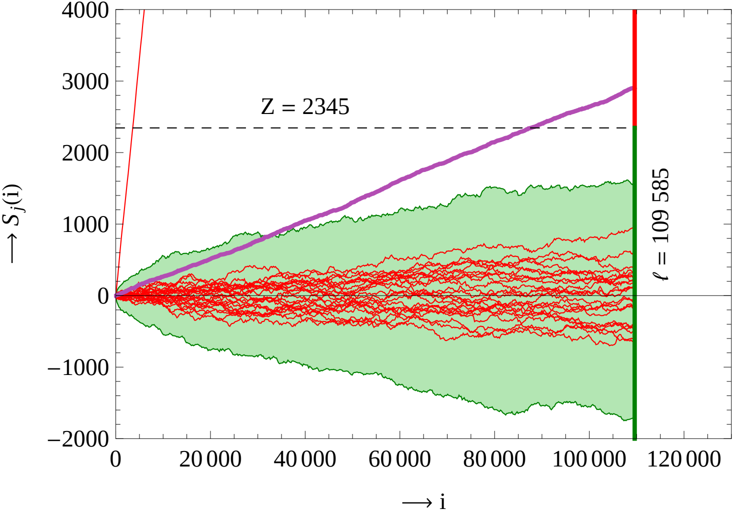
The static Tardos scheme

Catch many, sometimes.



The static Tardos scheme

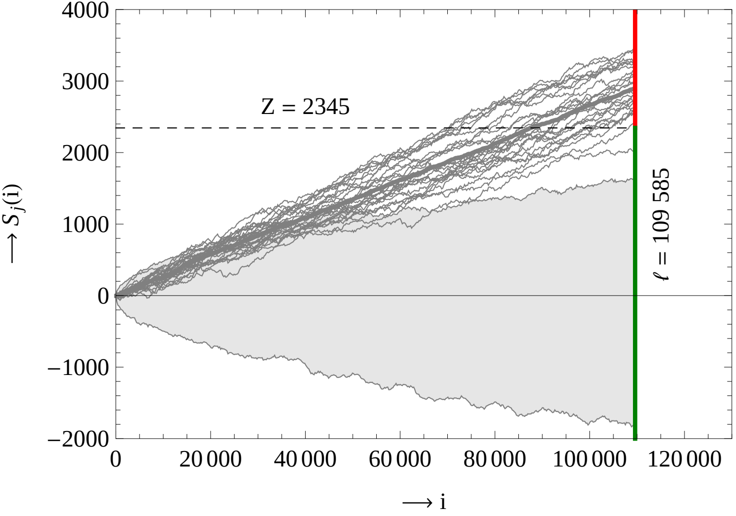
Catch many, sometimes. Catch one, worst-case.



TU/e

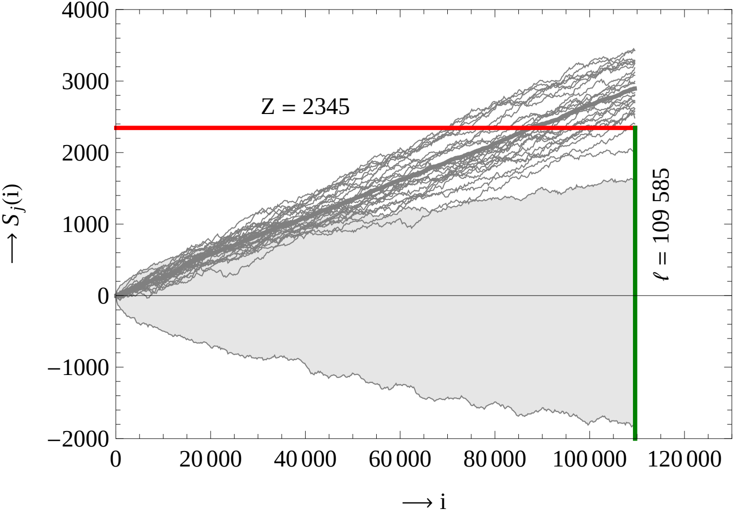
The static Tardos scheme

Instead of disconnecting pirates at the end...



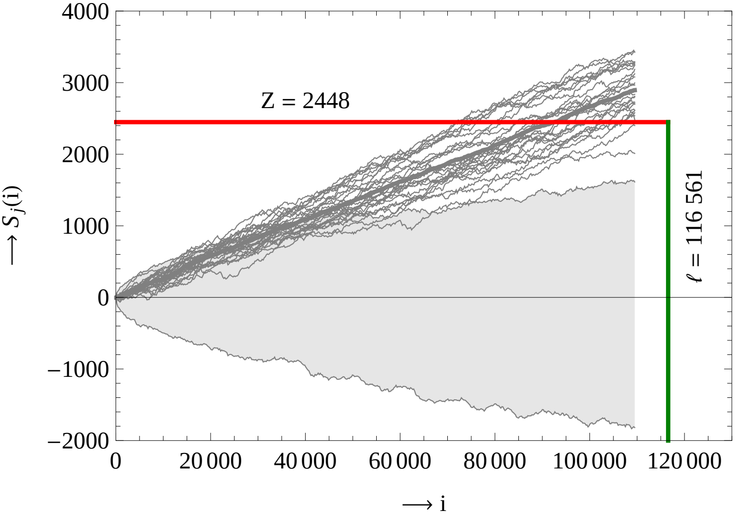
The dynamic Tardos scheme

Instead of disconnecting pirates at the end, disconnect them asap!



The dynamic Tardos scheme

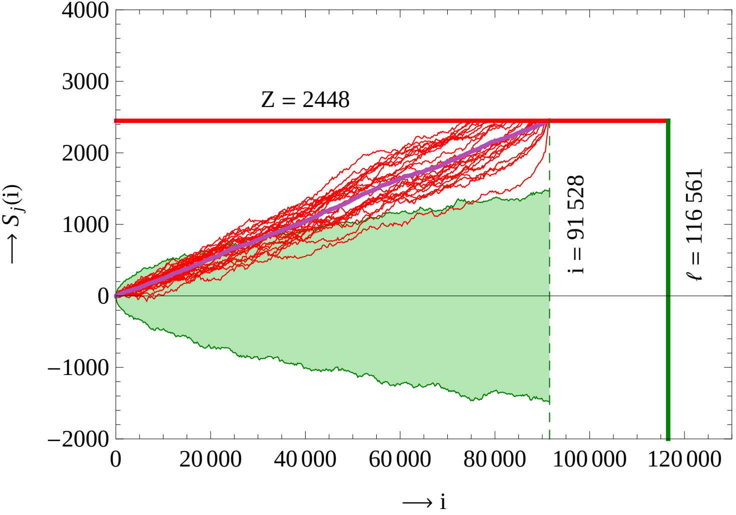
Instead of disconnecting pirates at the end, disconnect them asap!



TU/e

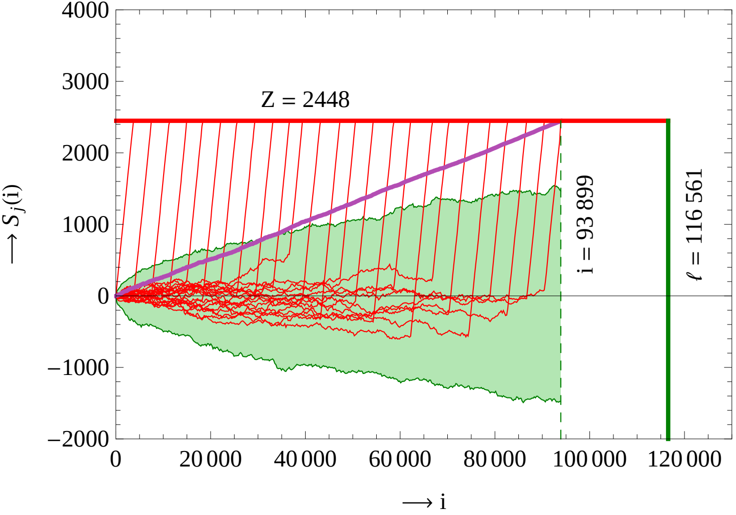
The dynamic Tardos scheme

Catch all, worst-case!



The dynamic Tardos scheme

Catch all, worst-case!



The dynamic Tardos scheme

Main result: By disconnecting users as soon as their scores are too large, we can catch *all pirates* in the worst-case with equivalent code lengths as in the static setting.

Static vs. Dynamic

Previous slides dealt with the *static setting*:

- First generate X, then y, then trace.
- Static content: video on demand, software, ...
- Optimized Tardos scheme achieves capacity.

In some practical scenarios, we have a *dynamic setting*:

- $X_1, y_1, X_2, y_2, \dots, X_\ell, y_\ell$.
- Dynamic content: live streams of cricket, football, ...
- More efficient schemes?
- What is the capacity?

Static vs. Dynamic

Previous slides dealt with the *static setting*:

- First generate X, then y, then trace.
- Static content: video on demand, software, ...
- Optimized Tardos scheme achieves capacity.

In some practical scenarios, we have a *dynamic setting*:

- $X_1, y_1, X_2, y_2, \dots, X_\ell, y_\ell$.
- Dynamic content: live streams of cricket, football, ...
- More efficient schemes? The dynamic Tardos scheme!
- What is the capacity?

Static vs. Dynamic

Previous slides dealt with the *static setting*:

- First generate X, then y, then trace.
- Static content: video on demand, software, ...
- Optimized Tardos scheme achieves capacity.

In some practical scenarios, we have a *dynamic setting*:

- $X_1, y_1, X_2, y_2, \ldots, X_\ell, y_\ell$.
- Dynamic content: live streams of cricket, football, ...
- More efficient schemes? The dynamic Tardos scheme!
- What is the capacity?

Problem: Blood testing

User								
Antonino	1	0	0	0	0	0	0	0
Boris	0	1	0	0	0	0	0	0
Caroline	0	0	1	0	0	0	0	0
David	0	0	0	1	0	0	0	0
Eve	0	0	0	0	1	0	0	0
Fred	0	0	0	0	0	1	0	0
Gábor	0	0	0	0	0	0	1	0
Henry	0	0	0	0	0	0	0	1

Problem: Blood testing

User								
Antonino	1	0	0	0	0	0	0	0
Boris	0	1	0	0	0	0	0	0
Caroline	0	0	1	0	0	0	0	0
David	0	0	0	1	0	0	0	0
Eve	0	0	0	0	1	0	0	0
Fred	0	0	0	0	0	1	0	0
Gábor	0	0	0	0	0	0	1	0
Henry	0	0	0	0	0	0	0	1

Problem: Blood testing

User								
Antonino	1	0	0	0	0	0	0	0
Boris	0	1	0	0	0	0	0	0
Caroline	0	0	1	0	0	0	0	0
David	0	0	0	1	0	0	0	0
Eve	0	0	0	0	1	0	0	0
Fred	0	0	0	0	0	1	0	0
Gábor	0	0	0	0	0	0	1	0
Henry	0	0	0	0	0	0	0	1
Result	0	0	1	0	0	0	0	0

Solution: Using pools

User								
Antonino	0	0	0					
Boris	0	0	1					
Caroline	0	1	0					
David	0	1	1					
Eve	1	0	0					
Fred	1	0	1					
Gábor	1	1	0					
Henry	1	1	1					

Solution: Using pools

User								
Antonino	0	0	0					
Boris	0	0	1					
Caroline	0	1	0					
David	0	1	1					
Eve	1	0	0					
Fred	1	0	1					
Gábor	1	1	0					
Henry	1	1	1					

Solution: Using pools

User								
Antonino	0	0	0					
Boris	0	0	1					
Caroline	0	1	0					
David	0	1	1					
Eve	1	0	0					
Fred	1	0	1					
Gábor	1	1	0					
Henry	1	1	1					
Result	0	1	0					

Problem: Several defectives

User								
Antonino	0	0	0					
Boris	0	0	1					
Caroline	0	1	0					
David	0	1	1					
Eve	1	0	0					
Fred	1	0	1					
Gábor	1	1	0					
Henry	1	1	1					

Problem: Several defectives

User								
Antonino	0	0	0					
Boris	0	0	1					
Caroline	0	1	0					
David	0	1	1					
Eve	1	0	0					
Fred	1	0	1					
Gábor	1	1	0					
Henry	1	1	1					

Problem: Several defectives

User								
Antonino	0	0	0					
Boris	0	0	1					
Caroline	0	1	0					
David	0	1	1					
Eve	1	0	0					
Fred	1	0	1					
Gábor	1	1	0					
Henry	1	1	1					
Result	1	1	0					

Traitor tracing: All-1 attack

User								
Antonino	0	0	0					
Boris	0	0	1					
Caroline	0	1	0					
David	0	1	1					
Eve	1	0	0					
Fred	1	0	1					
Gábor	1	1	0					
Henry	1	1	1					
Result	1	1	0					

The Tardos scheme

1. An algorithm to construct collusion-resistant codes
   1a. For each segment i, generate $p_i \sim F$.

$$F(p_i) = \tfrac{2}{\pi} \arcsin \sqrt{p_i}$$

   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 $S_{j,i} = g(X_{j,i}, y_i, p_i)$.

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

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

TU/e

A group testing scheme

1. An algorithm to construct collusion-resistant codes
   1a. For each segment i, generate $p_i \sim F$.

$$F(p_i) = \frac{2}{\pi} \arcsin \sqrt{p_i}$$

   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 $S_{j,i} = g(X_{j,i}, y_i, p_i)$.

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

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

A group testing scheme: Terminology

1. An algorithm to construct the group testing matrix
   - 1a. For each test i, generate $p_i \sim F$.

$$F(p_i) = \frac{2}{\pi} \arcsin \sqrt{p_i}.$$

   - 1b. For each test i, person j, choose $X_{j,i} = 1$ with prob. p_i.
2. An algorithm to trace test results to defectives
   - 2a. For each test i, person j, calculate $S_{j,i} = g(X_{j,i}, y_i, p_i)$.

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

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

A group testing scheme: Optimization

1. An algorithm to construct the group testing matrix
   - 1a. For each test i, generate $p_i \sim F$.

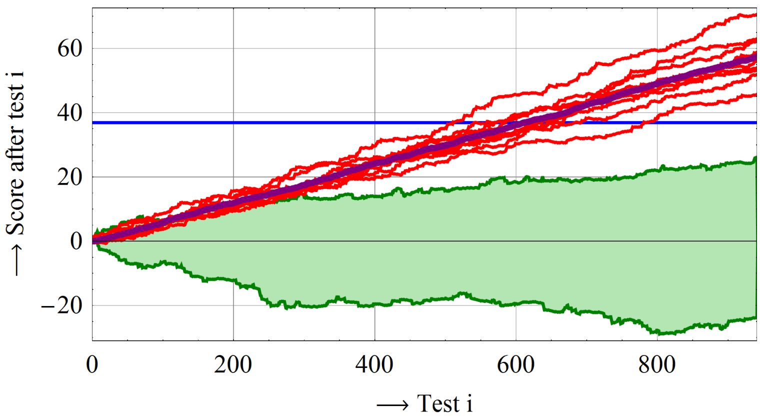
$$F(p_i) = H(p_i - p). \quad (p \approx \tfrac{1}{c})$$

   - 1b. For each test i, person j, choose $X_{j,i} = 1$ with prob. p_i.
2. An algorithm to trace test results to defectives
   - 2a. For each test i, person j, calculate $S_{j,i} = g(X_{j,i}, y_i, p_i)$.

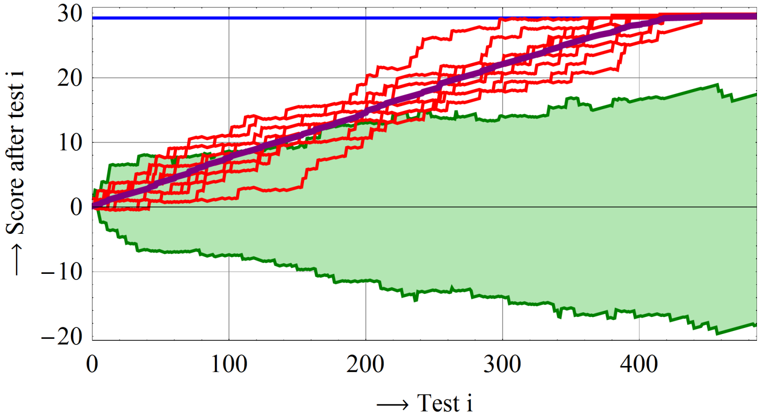
$$g(X_{j,i}, y_i, p_i) = \begin{cases} +p_i/(1-p_i), & \text{if } X_{j,i} = 0, y_i = 0, \\ -p_i(1-p_i)^{c-1}/(1-(1-p_i)^c), & \text{if } X_{j,i} = 0, y_i = 1, \\ -1, & \text{if } X_{j,i} = 1, y_i = 0, \\ +(1-p_i)^c/(1-(1-p_i)^c), & \text{if } X_{j,i} = 1, y_i = 1. \end{cases}$$

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

A group testing scheme: Static



A group testing scheme: Dynamic



Applications to group testing

First result: Using the score function $g = g^{\mathrm{all}-1}$ given by

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

and taking $p_i \equiv p \approx \frac{1}{c}$, we get a group testing scheme that uses

$$\ell \sim 2c \log n$$

group tests to find all c ill people with probability at least $1 - \varepsilon$.

Applications to group testing

Second result: By optimizing g and p, we get a framework for *arbitrary* group testing models:

- Noisy group testing, dilution: $\ell \sim 2c \log n/(1-r)$
- Noisy group testing, additive: $\ell \sim 2c \log n/(1-\sqrt{2r})$
- ...
- Threshold group testing, majority: $\ell \sim \pi c \log n$
- Threshold group testing, Bernoulli: $\ell \sim 4c \log n$
- Threshold group testing, linear gap: $\ell \sim 2c^2 \log n$
- ...

Applications to group testing

Second result: By optimizing g and p, we get a framework for *arbitrary* group testing models:

- Noisy group testing, dilution: $\ell \sim 2c \log n/(1-r)$
- Noisy group testing, additive: $\ell \sim 2c \log n/(1-\sqrt{2r})$
- ...
- Threshold group testing, majority: $\ell \sim \pi c \log n$
- Threshold group testing, Bernoulli: $\ell \sim 4c \log n$
- Threshold group testing, linear gap: $\ell \sim 2c^2 \log n$
- ...

We can combine this with the dynamic Tardos idea to get more efficient *adaptive* group testing schemes as well!

Conclusion

Progress in Traitor Tracing...

- Outline of the Tardos scheme
- A capacity-achieving score function
- A dynamic version of Tardos' scheme
- Open problem: Dynamic capacity?

...with Applications to Group Testing

- Construction based on the Tardos scheme
- Optimized bias (p) and score function
- Improves upon best known results
- Framework can be applied to other models
- Can be improved in the adaptive setting

Questions?