Nearest neighbor decoding for Tardos fingerprinting codes

Thijs Laarhoven

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

IH&MMSec 2019, Paris, France
(July 5, 2019)

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	…

TU/e

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	...

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	...

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	…

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		…
David	X	…
Eve		…
Fred		…
Gábor		…
Henry		…
Copy	y	…

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

Tardos' scheme

1. An algorithm to construct collusion-resistant codes

2. An algorithm to trace pirate copies to colluders

Tardos' scheme

1. An algorithm to construct collusion-resistant codes
   1a. For each segment i, generate $p_i \sim F$.
   1b. For each segment i, user j, choose $X_{j,i} = 1$ with probability p_i.
2. An algorithm to trace pirate copies to colluders

Tardos' scheme

1. An algorithm to construct collusion-resistant codes
   - 1a. For each segment i, generate $p_i \sim F$.
   - 1b. For each segment i, user j, choose $X_{j,i} = 1$ with probability p_i.
2. An algorithm to trace pirate copies to colluders
   - 2a. For each segment i, user j, calculate $S_{j,i} = g(X_{j,i}, y_i, p_i)$.
   - 2b. For each user j, accuse user j iff $S_j = \sum_i S_{j,i} > \eta$ is "large".

Tardos' scheme

p_i	p_1	p_2	p_3	p_4	p_5	$\cdots$	p_{1200}
Antonino	$X_{1,1}$	$X_{1,2}$	$X_{1,3}$	$X_{1,4}$	$X_{1,5}$	$\cdots$	$X_{1,1200}$
Boris	$X_{2,1}$	$X_{2,2}$	$X_{2,3}$	$X_{2,4}$	$X_{2,5}$	$\cdots$	$X_{2,1200}$
Caroline	$X_{3,1}$	$X_{3,2}$	$X_{3,3}$	$X_{3,4}$	$X_{3,5}$	$\cdots$	$X_{3,1200}$
David	$X_{4,1}$	$X_{4,2}$	$X_{4,3}$	$X_{4,4}$	$X_{4,5}$	$\cdots$	$X_{4,1200}$
Eve	$X_{5,1}$	$X_{5,2}$	$X_{5,3}$	$X_{5,4}$	$X_{5,5}$	$\cdots$	$X_{5,1200}$
Fred	$X_{6,1}$	$X_{6,2}$	$X_{6,3}$	$X_{6,4}$	$X_{6,5}$	$\cdots$	$X_{6,1200}$
Gábor	$X_{7,1}$	$X_{7,2}$	$X_{7,3}$	$X_{7,4}$	$X_{7,5}$	$\cdots$	$X_{7,1200}$
Henry	$X_{8,1}$	$X_{8,2}$	$X_{8,3}$	$X_{8,4}$	$X_{8,5}$	$\cdots$	$X_{8,1200}$
Copy	y_1	y_2	y_3	y_4	y_5	$\cdots$	y_{1200}

Tardos' scheme

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	$X_{1,1}$	$X_{1,2}$	$X_{1,3}$	$X_{1,4}$	$X_{1,5}$	$\cdots$	$X_{1,1200}$
Boris	$X_{2,1}$	$X_{2,2}$	$X_{2,3}$	$X_{2,4}$	$X_{2,5}$	$\cdots$	$X_{2,1200}$
Caroline	$X_{3,1}$	$X_{3,2}$	$X_{3,3}$	$X_{3,4}$	$X_{3,5}$	$\cdots$	$X_{3,1200}$
David	$X_{4,1}$	$X_{4,2}$	$X_{4,3}$	$X_{4,4}$	$X_{4,5}$	$\cdots$	$X_{4,1200}$
Eve	$X_{5,1}$	$X_{5,2}$	$X_{5,3}$	$X_{5,4}$	$X_{5,5}$	$\cdots$	$X_{5,1200}$
Fred	$X_{6,1}$	$X_{6,2}$	$X_{6,3}$	$X_{6,4}$	$X_{6,5}$	$\cdots$	$X_{6,1200}$
Gábor	$X_{7,1}$	$X_{7,2}$	$X_{7,3}$	$X_{7,4}$	$X_{7,5}$	$\cdots$	$X_{7,1200}$
Henry	$X_{8,1}$	$X_{8,2}$	$X_{8,3}$	$X_{8,4}$	$X_{8,5}$	$\cdots$	$X_{8,1200}$
Copy	y_1	y_2	y_3	y_4	y_5	$\cdots$	y_{1200}

Tardos' scheme

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	$X_{1,1}$	$X_{1,2}$	$X_{1,3}$	$X_{1,4}$	$X_{1,5}$	…	$X_{1,1200}$
Boris	$X_{2,1}$	$X_{2,2}$	$X_{2,3}$	$X_{2,4}$	$X_{2,5}$	…	$X_{2,1200}$
Caroline	$X_{3,1}$	$X_{3,2}$	$X_{3,3}$	$X_{3,4}$	$X_{3,5}$	…	$X_{3,1200}$
David	$X_{4,1}$	$X_{4,2}$	$X_{4,3}$	$X_{4,4}$	$X_{4,5}$	…	$X_{4,1200}$
Eve	$X_{5,1}$	$X_{5,2}$	$X_{5,3}$	$X_{5,4}$	$X_{5,5}$	…	$X_{5,1200}$
Fred	$X_{6,1}$	$X_{6,2}$	$X_{6,3}$	$X_{6,4}$	$X_{6,5}$	…	$X_{6,1200}$
Gábor	$X_{7,1}$	$X_{7,2}$	$X_{7,3}$	$X_{7,4}$	$X_{7,5}$	…	$X_{7,1200}$
Henry	$X_{8,1}$	$X_{8,2}$	$X_{8,3}$	$X_{8,4}$	$X_{8,5}$	…	$X_{8,1200}$
Copy	y_1	y_2	y_3	y_4	y_5	…	y_{1200}

Tardos' scheme

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	$X_{1,1}$	$X_{1,2}$	$X_{1,3}$	$X_{1,4}$	$X_{1,5}$	…	$X_{1,1200}$
Boris	$X_{2,1}$	$X_{2,2}$	$X_{2,3}$	$X_{2,4}$	$X_{2,5}$	…	$X_{2,1200}$
Caroline	$X_{3,1}$	$X_{3,2}$	$X_{3,3}$	$X_{3,4}$	$X_{3,5}$	…	$X_{3,1200}$
David	$X_{4,1}$	$X_{4,2}$	$X_{4,3}$	$X_{4,4}$	$X_{4,5}$	…	$X_{4,1200}$
Eve	$X_{5,1}$	$X_{5,2}$	$X_{5,3}$	$X_{5,4}$	$X_{5,5}$	…	$X_{5,1200}$
Fred	$X_{6,1}$	$X_{6,2}$	$X_{6,3}$	$X_{6,4}$	$X_{6,5}$	…	$X_{6,1200}$
Gábor	$X_{7,1}$	$X_{7,2}$	$X_{7,3}$	$X_{7,4}$	$X_{7,5}$	…	$X_{7,1200}$
Henry	$X_{8,1}$	$X_{8,2}$	$X_{8,3}$	$X_{8,4}$	$X_{8,5}$	…	$X_{8,1200}$
Copy	y_1	y_2	y_3	y_4	y_5	…	y_{1200}

Tardos' scheme

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_{1200}

Tardos' scheme

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

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

Tardos' scheme

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

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

Tardos' scheme

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}

Tardos' scheme

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	

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

Tardos' scheme

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	

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

Tardos' scheme

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

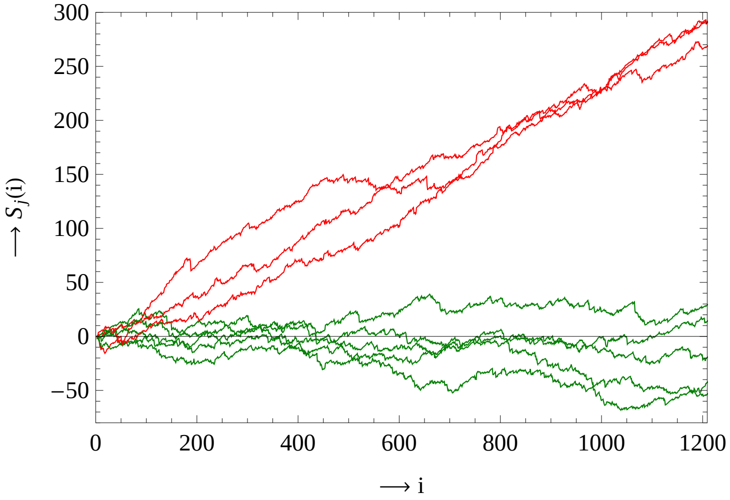
p_i	0.20	0.05	0.88	0.79	0.98	…	0.18	$\sum_i S_{j,i}$
Antonino	+0.5	+0.2	−0.4	+0.5	+0.1	…	+0.5	+14
Boris	−2.0	+0.2	−0.4	+0.5	+0.1	…	−2.1	−19
Caroline	−2.0	+0.2	+2.7	+0.5	−7.2	…	+0.5	+291
David	+0.5	+0.2	−0.4	+0.5	+0.1	…	+0.5	+29
Eve	+0.5	+0.2	−0.4	−1.9	+0.1	…	+0.5	+292
Fred	−2.0	+0.2	−0.4	−1.9	+0.1	…	+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}

Accused = {Caroline, Eve, Henry}

Tardos' scheme

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



Tardos' scheme

1. An algorithm to construct collusion-resistant codes
   1a. For each segment i, generate $p_i \sim F$.
   1b. For each segment i, user j, choose $X_{j,i} = 1$ with probability p_i.
2. An algorithm to trace pirate copies to colluders
   2a. For each segment i, user j, calculate $S_{j,i} = g(X_{j,i}, y_i, p_i)$.
   2b. For each user j, accuse user j iff $S_j = \sum_i S_{j,i} > \eta$ is "large".

Tardos' scheme

Bias distribution

- Tardos *(STOC 2003)*: arcsine distribution
- Nuida–Hagiwara–Watanabe–Imai *(IH 2007)*: optimal discrete shape (small c)
- Furon–Guyader–Cerou *(IH 2008)*: why the arcsine distribution
- Huang–Moulin *(WIFS 2010)*: arcsine distribution optimal (large c)
- Laarhoven–De Weger *(IH&MMSec 2014)*: optimal discrete distributions (small c) converge to arcsine distribution (large c)

$$f(p) = \frac{1}{\pi\sqrt{p(1-p)}}, \qquad p \in (0,1). \tag{1}$$

Tardos' scheme

Score functions

- Tardos *(STOC 2003)*: "asymmetric" score function
- Skoric–Katzenbeisser–Celik *(DCC 2008)*: symmetric score function
- Furon–Guyader–Cerou *(IH 2008)*: why these score functions
- Furon–Perez-Freire *(MMSec 2009)*: EM decoder
- Meerwald–Furon *(IEEE-TIFS 2012)*: iterative joint decoder
- Furon–Guyader–Cerou *(WIFS 2012)*: MCMC joint decoder
- Oosterwijk–Skoric–Doumen *(IH&MMSec 2013)*: optimal simple decoder
- Desoubeaux–Herzet–Puech–Guelvouit *(MMSP 2013)*: MAP-based joint decoder
- Laarhoven *(IH&MMSec 2014)*: more optimal simple decoders
- Furon–Desoubeaux *(WIFS 2014)*: comprehensive comparison of decoders

Tardos' scheme

Code lengths

- Tardos *(STOC 2003)*: $\ell = 100c^2 \ln n$
- Skoric–Vladimirova–Celik–Talstra *(IEEE-TIT 2006)*: $\ell \approx 39.48c^2 \ln n$
- Blayer–Tassa *(DCC 2008)*: $\ell \approx 19.74c^2 \ln n$
- Skoric–Katzenbeisser–Celik *(DCC 2008)*: $\ell \approx 9.87c^2 \ln n$
- Nuida–…–Imai *(DCC 2009)*: $\ell \approx 5.35c^2 \ln n$
- Laarhoven–De Weger *(DCC 2014)*: $\ell \approx 4.93c^2 \ln n$

- Nuida–Hagiwara–Watanabe–Imai *(IH 2007)*: code lengths for small collusions
- Amiri–Tardos *(SODA 2009)*: achievable code length with joint decoding
- Furon–Perez-Freire–Guyader–Cerou *(IH 2009)*: estimating ℓ in practice
- Berchtold–Schafer *(MMSec 2012)*: optimizing ℓ for joint decoders

Tardos' scheme

Decoding methods

- Tardos *(STOC 2003)*: linear-time simple decoder
- Amiri–Tardos *(SODA 2009)*: theoretical joint decoder
- Furon–Perez-Freire *(MMSec 2009)*: fast EM decoder
- Laarhoven–Doumen–Roelse–Skoric–De Weger *(IEEE-TIT 2011)*: dynamic decoder
- Meerwald–Furon *(IEEE-TIFS 2012)*: iterative joint decoder
- Furon–Guyader–Cerou *(WIFS 2012)*: MCMC joint decoder
- Desoubeaux–Herzet–Puech–Guelvouit *(MMSP 2013)*: MAP-based joint decoder
- Laarhoven *(IH&MMSec 2015)*: sequential decoders

Tardos' scheme

Decoding methods

- Tardos *(STOC 2003)*: linear-time simple decoder
- Amiri–Tardos *(SODA 2009)*: theoretical joint decoder
- Furon–Perez-Freire *(MMSec 2009)*: fast EM decoder
- Laarhoven–Doumen–Roelse–Skoric–De Weger *(IEEE-TIT 2011)*: dynamic decoder
- Meerwald–Furon *(IEEE-TIFS 2012)*: iterative joint decoder
- Furon–Guyader–Cerou *(WIFS 2012)*: MCMC joint decoder
- Desoubeaux–Herzet–Puech–Guelvouit *(MMSP 2013)*: MAP-based joint decoder
- Laarhoven *(IH&MMSec 2015)*: sequential decoders
- **This work**: sublinear-time simple decoder

Nearest neighbor decoder

Main ideas

Intuition: View code words as high-dimensional vectors

- Code words, pirate output can be seen as length-ℓ binary vectors
- Usually colluder vectors have a higher similarity with pirate output
- Does not work for e.g. minority voting attack

Nearest neighbor decoder

Main ideas

Intuition: View code words as high-dimensional vectors

- Code words, pirate output can be seen as length-ℓ binary vectors
- Usually colluder vectors have a higher similarity with pirate output
- Does not work for e.g. minority voting attack

Actual scheme: View score vectors as high-dimensional vectors

- For each coordinate, two possible values depending on p_i
- Collusion strategy influences average scores
- Regardless of attack, colluder vectors most similar to output vector
- Can use nearest neighbor data structures for fast lookups

Nearest neighbor decoder

Algorithms

Preprocessing algorithm

- Create many hash buckets of similar score vectors
- Store all user score vectors in these hash buckets
  - Limitation: Only seems to work for symmetric score function

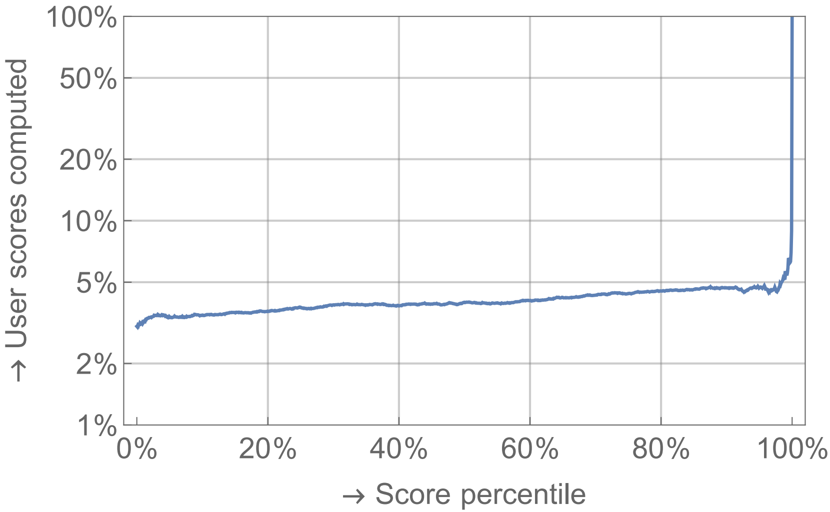
Tracing algorithm

- Given pirate output, find buckets which contain similar vectors
- Compute scores only for users in these hash buckets

Generally: need more space, but find colluders faster

Nearest neighbor decoder

Experimental data



Nearest neighbor decoder

Time–space trade-offs

c	linear space	balanced trade-off	instant decoding
1	$(S,T)=(n,\log n)$		
2	$(S,T)=(n,n^{0.75})$	$(S,T)=(n^{1.33},n^{0.33})$	$(S,T)=(n^{5.00},n^{o(1)})$
3	$(S,T)=(n,n^{0.89})$	$(S,T)=(n^{1.50},n^{0.50})$	$(S,T)=(n^{8.00},n^{o(1)})$
4	$(S,T)=(n,n^{0.94})$	$(S,T)=(n^{1.60},n^{0.60})$	$(S,T)=(n^{15.0},n^{o(1)})$
5	$(S,T)=(n,n^{0.96})$	$(S,T)=(n^{1.68},n^{0.68})$	$(S,T)=(n^{25.8},n^{o(1)})$
6	$(S,T)=(n,n^{0.97})$	$(S,T)=(n^{1.72},n^{0.72})$	$(S,T)=(n^{37.6},n^{o(1)})$
7	$(S,T)=(n,n^{0.98})$	$(S,T)=(n^{1.77},n^{0.77})$	$(S,T)=(n^{57.3},n^{o(1)})$
8	$(S,T)=(n,n^{0.99})$	$(S,T)=(n^{1.79},n^{0.79})$	$(S,T)=(n^{75.2},n^{o(1)})$

Main conclusions

Good news

- Can significantly decrease the decoding time in practice
- Can be used in place, with exact same code constructions
- May be useful for online/live applications with quick decisions

Main conclusions

Good news

- Can significantly decrease the decoding time in practice
- Can be used in place, with exact same code constructions
- May be useful for online/live applications with quick decisions

Bad news

- Does not scale well with the collusion size
- Improvement not big enough to make joint decoding more practical
- Does not work nicely for other score functions
- Decoding time is not the main bottleneck in practice

Main conclusions

Good news

- Can significantly decrease the decoding time in practice
- Can be used in place, with exact same code constructions
- May be useful for online/live applications with quick decisions

Bad news

- Does not scale well with the collusion size
- Improvement not big enough to make joint decoding more practical
- Does not work nicely for other score functions
- Decoding time is not the main bottleneck in practice

Bon appétit!