Discrete Distributions in the Tardos Scheme, Revisited

Thijs Laarhoven, Benne de Weger

t.m.m.laarhoven@tue.nl
http://www.thijs.com/

EiPSI Seminar, Eindhoven, The Netherlands
(May 7, 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	...
Chris	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	...
Chris	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	...
Chris	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	...
Chris	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	...
Chris	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	...
Chris	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	...
Chris	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	...
Chris	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	...
Chris	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	...
Chris	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	...

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	...
Chris	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	...

1. An algorithm to construct collusion-resistant codes

TU/e

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	...
Chris	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	...

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 $p_i \sim F$.

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 $p_i \sim F$.
   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

The Tardos scheme: Overview

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 prob. p_i.
2. An algorithm to trace pirate copies to colluders
   - 2a. For each segment i, user j, calculate $S_{j,i} = g(X_{j,i}, y_i, p_i)$.

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

The Tardos scheme: Overview

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 prob. p_i.
2. An algorithm to trace pirate copies to colluders
   2a. For each segment i, user j, calculate $S_{j,i} = g(X_{j,i}, y_i, p_i)$.

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

   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	$X_{1,1}$	$X_{1,2}$	$X_{1,3}$	$X_{1,4}$	$X_{1,5}$	$\cdots$	$X_{1,1200}$
Boris	$X_{2,1}$	$X_{2,2}$	$X_{2,3}$	$X_{2,4}$	$X_{2,5}$	$\cdots$	$X_{2,1200}$
Chris	$X_{3,1}$	$X_{3,2}$	$X_{3,3}$	$X_{3,4}$	$X_{3,5}$	$\cdots$	$X_{3,1200}$
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}

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	$X_{1,1}$	$X_{1,2}$	$X_{1,3}$	$X_{1,4}$	$X_{1,5}$	$\cdots$	$X_{1,1200}$
Boris	$X_{2,1}$	$X_{2,2}$	$X_{2,3}$	$X_{2,4}$	$X_{2,5}$	$\cdots$	$X_{2,1200}$
Chris	$X_{3,1}$	$X_{3,2}$	$X_{3,3}$	$X_{3,4}$	$X_{3,5}$	$\cdots$	$X_{3,1200}$
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}

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	$X_{1,1}$	$X_{1,2}$	$X_{1,3}$	$X_{1,4}$	$X_{1,5}$	...	$X_{1,1200}$
Boris	$X_{2,1}$	$X_{2,2}$	$X_{2,3}$	$X_{2,4}$	$X_{2,5}$	...	$X_{2,1200}$
Chris	$X_{3,1}$	$X_{3,2}$	$X_{3,3}$	$X_{3,4}$	$X_{3,5}$	...	$X_{3,1200}$
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}

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	$\dots$	0.18
Antonino	$X_{1,1}$	$X_{1,2}$	$X_{1,3}$	$X_{1,4}$	$X_{1,5}$	$\dots$	$X_{1,1200}$
Boris	$X_{2,1}$	$X_{2,2}$	$X_{2,3}$	$X_{2,4}$	$X_{2,5}$	$\dots$	$X_{2,1200}$
Chris	$X_{3,1}$	$X_{3,2}$	$X_{3,3}$	$X_{3,4}$	$X_{3,5}$	$\dots$	$X_{3,1200}$
David	$X_{4,1}$	$X_{4,2}$	$X_{4,3}$	$X_{4,4}$	$X_{4,5}$	$\dots$	$X_{4,1200}$
Eve	$X_{5,1}$	$X_{5,2}$	$X_{5,3}$	$X_{5,4}$	$X_{5,5}$	$\dots$	$X_{5,1200}$
Fred	$X_{6,1}$	$X_{6,2}$	$X_{6,3}$	$X_{6,4}$	$X_{6,5}$	$\dots$	$X_{6,1200}$
Gábor	$X_{7,1}$	$X_{7,2}$	$X_{7,3}$	$X_{7,4}$	$X_{7,5}$	$\dots$	$X_{7,1200}$
Henry	$X_{8,1}$	$X_{8,2}$	$X_{8,3}$	$X_{8,4}$	$X_{8,5}$	$\dots$	$X_{8,1200}$
Copy	y_1	y_2	y_3	y_4	y_5	$\dots$	y_{1200}

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
Chris	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	0	0	0	1	1	...	0
Copy	y_1	y_2	y_3	y_4	y_5	...	y_{1200}

The Tardos scheme: Coalition

Pirates get their versions, ...

p_i	.	.	.	.	.	...	.
Antonino	.	.	.	.	.	...	.
Boris	.	.	.	.	.	...	.
Chris	1	0	0	1	0	...	0
David	.	.	.	.	.	...	.
Eve	0	0	1	0	1	...	0
Fred	.	.	.	.	.	...	.
Gábor	.	.	.	.	.	...	.
Henry	0	0	0	1	1	...	0
Copy	y_1	y_2	y_3	y_4	y_5	...	y_{1200}

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

The Tardos scheme: Coalition

Pirates get their versions, compare them ...

p_i	.	.	.	.	.	...	.
Antonino	.	.	.	.	.	...	.
Boris	.	.	.	.	.	...	.
Chris	**1**	0	**0**	**1**	**0**	...	0
David	.	.	.	.	.	...	.
Eve	**0**	0	**1**	**0**	**1**	...	0
Fred	.	.	.	.	.	...	.
Gábor	.	.	.	.	.	...	.
Henry	**0**	0	**0**	**1**	**1**	...	0
Copy	y_1	y_2	y_3	y_4	y_5	...	y_{1200}

Coalition = {Chris, Eve, Henry}

The Tardos scheme: Coalition

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

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

Coalition $= \{$Chris, 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
Chris	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	0	0	0	1	1	...	0
Copy	0	0	0	1	1	...	0

Coalition $=$ {Chris, 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
Chris	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	0	0	0	1	1	...	0
Copy	0	0	0	1	1	...	0

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

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
Chris	-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	$+0.5$	$+0.2$	$+2.7$	$+0.5$	$+0.1$	...	$+0.5$
Copy	0	0	0	1	1	...	0

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

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	S_j
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
Chris	−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	+0.5	+0.2	+2.7	+0.5	+0.1	...	+0.5	0
Copy	0	0	0	1	1	...	0	

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

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	S_j
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
Chris	-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	$+0.5$	$+0.2$	$+2.7$	$+0.5$	$+0.1$	...	$+0.5$	$+269$
Copy	0	0	0	1	1	...	0	

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

TU/e

The Tardos scheme: Scores

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

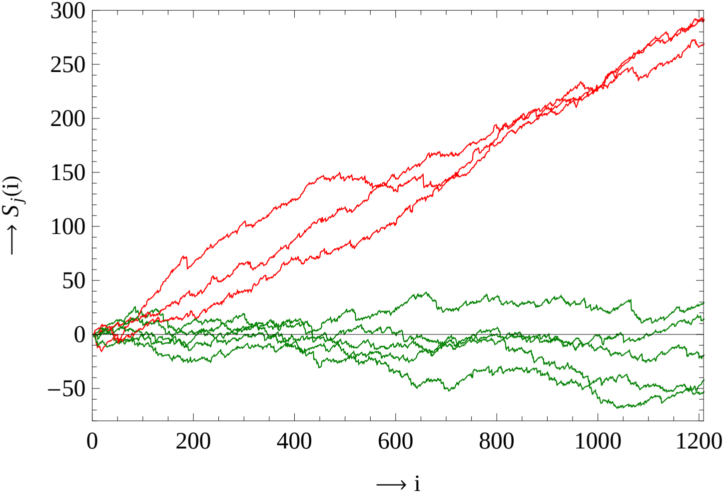
p_i	0.20	0.05	0.88	0.79	0.98	…	0.18	S_j
Antonino	$+0.5$	$+0.2$	-0.4	$+0.5$	$+0.1$	…	$+0.5$	$+14$
Boris	-2.0	$+0.2$	-0.4	$+0.5$	$+0.1$	…	-2.1	-19
Chris	-2.0	$+0.2$	$+2.7$	$+0.5$	-7.2	…	$+0.5$	$+291$
David	$+0.5$	$+0.2$	-0.4	$+0.5$	$+0.1$	…	$+0.5$	$+29$
Eve	$+0.5$	$+0.2$	-0.4	-1.9	$+0.1$	…	$+0.5$	$+292$
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	$+0.5$	$+0.2$	$+2.7$	$+0.5$	$+0.1$	…	$+0.5$	$+269$
Copy	0	0	0	1	1	…	0	

Coalition $=$ {Chris, Eve, Henry}

Accused $=$ {Chris, Eve, Henry}

The Tardos scheme: Scores

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 $p_i \sim F$.
   - 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 $S_{j,i} = g(X_{j,i}, y_i, p_i)$.

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

   - 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 $p_i \sim F$.
   - 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 $S_{j,i} = g(X_{j,i}, y_i, p_i)$.

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

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

How to choose F?

How to choose F

How to choose F

- Continuous distributions

- Discrete distributions

How to choose F

- Continuous distributions
  - Arcsine distributions with cutoffs

- Discrete distributions

How to choose F

- Continuous distributions
  - Arcsine distributions with cutoffs
  - Allows proof via Markov's inequality

- Discrete distributions

How to choose F

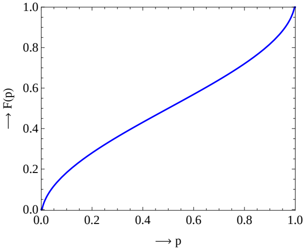
- Continuous distributions
  - Arcsine distributions with cutoffs
  - Allows proof via Markov's inequality
  - Number of segments needed:
    - Small c: About $10c^2 \ln(n/\varepsilon_1)$
    - Large c: Converges to $4.93c^2 \ln(n/\varepsilon_1)$
- Discrete distributions

How to choose F

- Continuous distributions
  - Arcsine distributions with cutoffs
  - Allows proof via Markov's inequality
  - Number of segments needed:
    - Small c: About $10c^2 \ln(n/\varepsilon_1)$
    - Large c: Converges to $4.93c^2 \ln(n/\varepsilon_1)$
  - Converges to arcsine distribution
- Discrete distributions

How to choose F

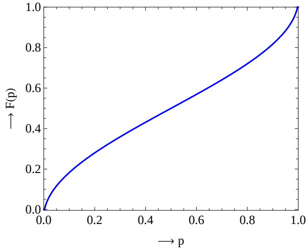
- Continuous distributions
  - Arcsine distributions with cutoffs
  - Allows proof via Markov's inequality
  - Number of segments needed:
    - Small c: About $10c^2 \ln(n/\varepsilon_1)$
    - Large c: Converges to $4.93c^2 \ln(n/\varepsilon_1)$
  - Converges to arcsine distribution
- Discrete distributions



TU/e

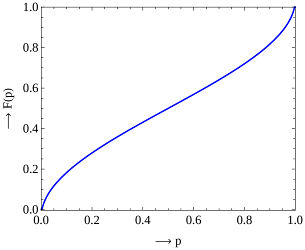
How to choose F

- Continuous distributions
  - Arcsine distributions with cutoffs
  - Allows proof via Markov's inequality
  - Number of segments needed:
    - Small c: About $10c^2 \ln(n/\varepsilon_1)$
    - Large c: Converges to $4.93c^2 \ln(n/\varepsilon_1)$
  - Converges to arcsine distribution
- Discrete distributions
  - Based on Gauss-Legendre quadratures



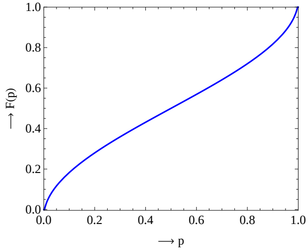
How to choose F

- Continuous distributions
  - Arcsine distributions with cutoffs
  - Allows proof via Markov's inequality
  - Number of segments needed:
    - Small c: About $10c^2 \ln(n/\varepsilon_1)$
    - Large c: Converges to $4.93c^2 \ln(n/\varepsilon_1)$
  - Converges to arcsine distribution
- Discrete distributions
  - Based on Gauss-Legendre quadratures
  - Maximizes the expected coalition score



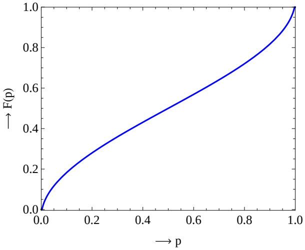
How to choose F

- Continuous distributions
  - Arcsine distributions with cutoffs
  - Allows proof via Markov's inequality
  - Number of segments needed:
    - Small c: About $10c^2 \ln(n/\varepsilon_1)$
    - Large c: Converges to $4.93c^2 \ln(n/\varepsilon_1)$
  - Converges to arcsine distribution
- Discrete distributions
  - Based on Gauss-Legendre quadratures
  - Maximizes the expected coalition score
  - Number of segments needed:
    - Small c: About $4c^2 \ln(n/\varepsilon_1)$
    - Large c: About $5.35c^2 \ln(n/\varepsilon_1)$



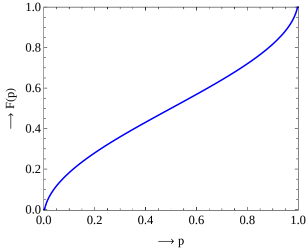
How to choose F

- Continuous distributions
  - Arcsine distributions with cutoffs
  - Allows proof via Markov's inequality
  - Number of segments needed:
    - Small c: About $10c^2 \ln(n/\varepsilon_1)$
    - Large c: Converges to $4.93c^2 \ln(n/\varepsilon_1)$
  - Converges to arcsine distribution
- Discrete distributions
  - Based on Gauss-Legendre quadratures
  - Maximizes the expected coalition score
  - Number of segments needed:
    - Small c: About $4c^2 \ln(n/\varepsilon_1)$
    - Large c: About $5.35c^2 \ln(n/\varepsilon_1)$
  - Converges to?

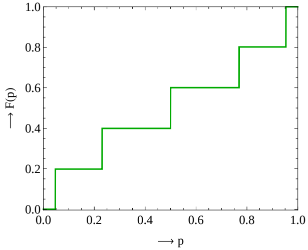


How to choose F

- Continuous distributions
  - Arcsine distributions with cutoffs
  - Allows proof via Markov's inequality
  - Number of segments needed:
    - Small c: About $10c^2 \ln(n/\varepsilon_1)$
    - Large c: Converges to $4.93c^2 \ln(n/\varepsilon_1)$
  - Converges to arcsine distribution

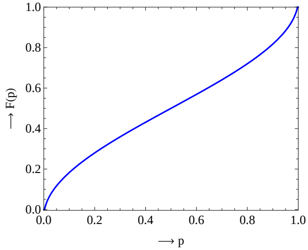


- Discrete distributions
  - Based on Gauss-Legendre quadratures
  - Maximizes the expected coalition score
  - Number of segments needed:
    - Small c: About $4c^2 \ln(n/\varepsilon_1)$
    - Large c: About $5.35c^2 \ln(n/\varepsilon_1)$
  - Converges to?

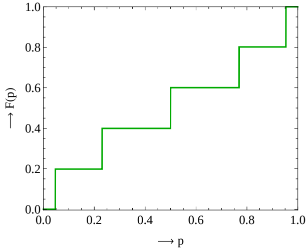


How to choose F

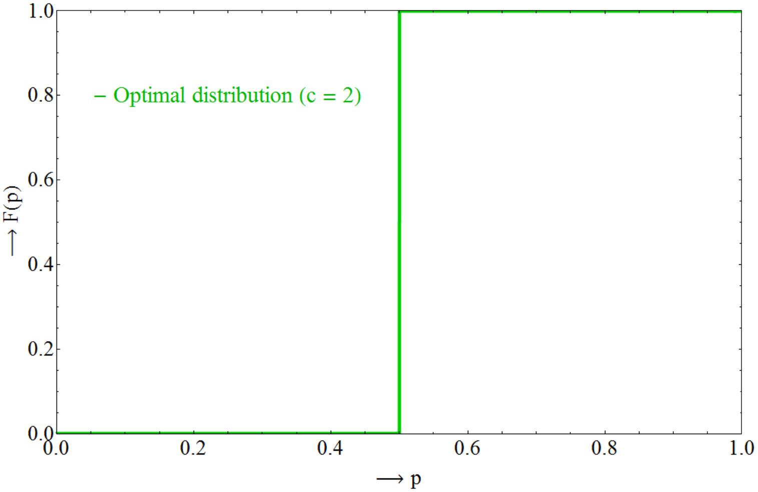
- Continuous distributions
  - Arcsine distributions with cutoffs
  - Allows proof via Markov's inequality
  - Number of segments needed:
    - Small c: About $10c^2 \ln(n/\varepsilon_1)$
    - Large c: Converges to $4.93c^2 \ln(n/\varepsilon_1)$
  - Converges to arcsine distribution



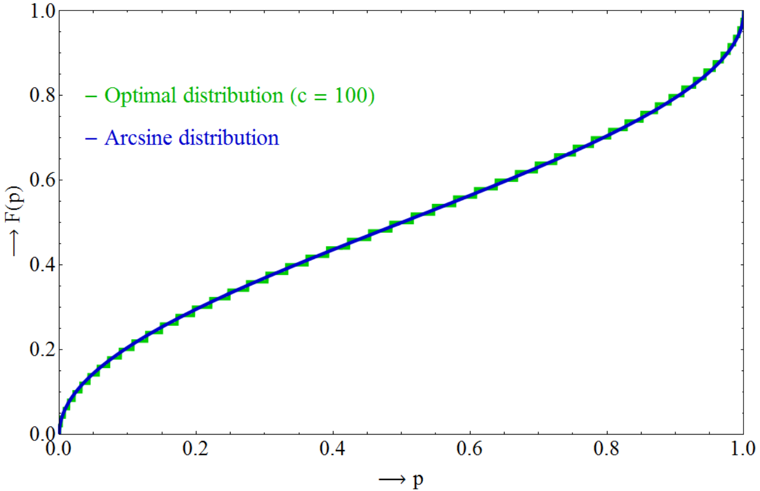
- Discrete distributions
  - Based on Gauss-Legendre quadratures
  - Maximizes the expected coalition score
  - Number of segments needed:
    - Small c: About $4c^2 \ln(n/\varepsilon_1)$
    - Large c: About $5.35c^2 \ln(n/\varepsilon_1)$
  - **Converges to?**



Discrete distributions

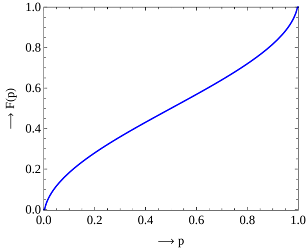


Discrete distributions

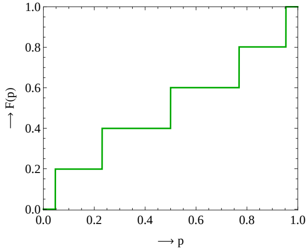


How to choose F

- Continuous distributions
  - Arcsine distribution with cutoffs
  - Allows proof via Markov's inequality
  - Number of segments needed:
    - Small c: About $10c^2 \ln(n/\varepsilon_1)$
    - Large c: Converges to $4.93c^2 \ln(n/\varepsilon_1)$
  - Converges to arcsine distribution

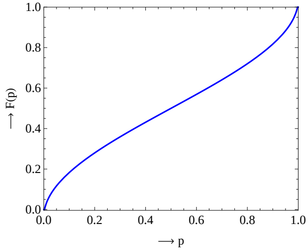


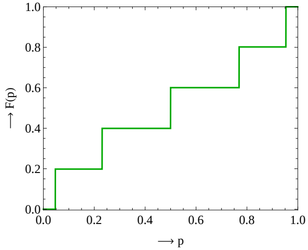
- Discrete distributions
  - Based on Gauss-Legendre quadratures
  - Maximizes the expected coalition score
  - Number of segments needed:
    - Small c: About $4c^2 \ln(n/\varepsilon_1)$
    - Large c: About $5.35c^2 \ln(n/\varepsilon_1)$
  - **Converges to?**



How to choose F

- Continuous distributions
  - Arcsine distribution with cutoffs
  - Allows proof via Markov's inequality
  - Number of segments needed:
    - Small c: About $10c^2 \ln(n/\varepsilon_1)$
    - Large c: Converges to $4.93c^2 \ln(n/\varepsilon_1)$
  - Converges to arcsine distribution
- Discrete distributions
  - Based on Gauss-Legendre quadratures
  - Maximizes the expected coalition score
  - Number of segments needed:
    - Small c: About $4c^2 \ln(n/\varepsilon_1)$
    - Large c: About $5.35c^2 \ln(n/\varepsilon_1)$
  - **Converges to arcsine distribution!**





Our contributions

Theorem: Discrete distributions converge to arcsine distribution

Our contributions

Theorem: Discrete distributions converge to arcsine distribution

- Proof: See our paper (bit technical)

Our contributions

Theorem: Discrete distributions converge to arcsine distribution

- Proof: See our paper (bit technical)
- Corollary: Arcsine distribution is asymptotically optimal

Our contributions

Theorem: Discrete distributions converge to arcsine distribution

- Proof: See our paper (bit technical)
- Corollary: Arcsine distribution is asymptotically optimal
- Corollary: Code length $4.93c^2 \ln(n/\varepsilon_1)$ is asympt. optimal

Our contributions

Theorem: Discrete distributions converge to arcsine distribution

- Proof: See our paper (bit technical)
- Corollary: Arcsine distribution is asymptotically optimal
- Corollary: Code length $4.93c^2 \ln(n/\varepsilon_1)$ is asympt. optimal
- Discrete and continuous distributions are not that different

Our contributions

Theorem: Discrete distributions converge to arcsine distribution

- Proof: See our paper (bit technical)
- Corollary: Arcsine distribution is asymptotically optimal
- Corollary: Code length $4.93c^2 \ln(n/\varepsilon_1)$ is asympt. optimal
- Discrete and continuous distributions are not that different

Construction: A practical alternative to the optimal distributions

Our contributions

Theorem: Discrete distributions converge to arcsine distribution

- Proof: See our paper (bit technical)
- Corollary: Arcsine distribution is asymptotically optimal
- Corollary: Code length $4.93c^2 \ln(n/\varepsilon_1)$ is asympt. optimal
- Discrete and continuous distributions are not that different

Construction: A practical alternative to the optimal distributions

- Approximations of the optimal distributions

Our contributions

Theorem: Discrete distributions converge to arcsine distribution

- Proof: See our paper (bit technical)
- Corollary: Arcsine distribution is asymptotically optimal
- Corollary: Code length $4.93c^2 \ln(n/\varepsilon_1)$ is asympt. optimal
- Discrete and continuous distributions are not that different

Construction: A practical alternative to the optimal distributions

- Approximations of the optimal distributions
- Simpler bias generation, calculations

Our contributions

Theorem: Discrete distributions converge to arcsine distribution

- Proof: See our paper (bit technical)
- Corollary: Arcsine distribution is asymptotically optimal
- Corollary: Code length $4.93c^2 \ln(n/\varepsilon_1)$ is asympt. optimal
- Discrete and continuous distributions are not that different

Construction: A practical alternative to the optimal distributions

- Approximations of the optimal distributions
- Simpler bias generation, calculations
- Same asymptotic behavior

TU/e

Our contributions

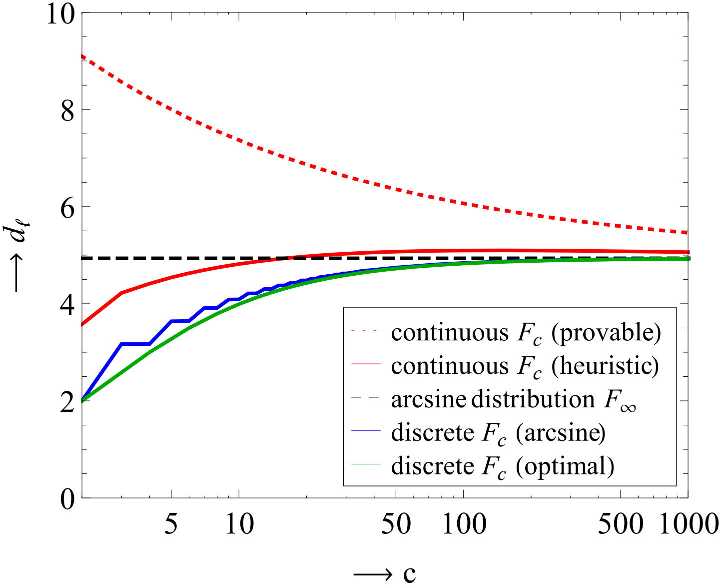
Theorem: Discrete distributions converge to arcsine distribution

- Proof: See our paper (bit technical)
- Corollary: Arcsine distribution is asymptotically optimal
- Corollary: Code length $4.93c^2 \ln(n/\varepsilon_1)$ is asympt. optimal
- Discrete and continuous distributions are not that different

Construction: A practical alternative to the optimal distributions

- Approximations of the optimal distributions
- Simpler bias generation, calculations
- Same asymptotic behavior
- Heuristics: Comparable performance

Comparison



10
8
6
4
2
0
$\longrightarrow d_\ell$
5
10
50
100
500
1000
$\longrightarrow$ c
continuous F_c (provable)
continuous F_c (heuristic)
arcsine distribution F_∞
discrete F_c (arcsine)
discrete F_c (optimal)

Questions?