Collusion-resistant fingerprinting schemes

Thijs Laarhoven

Technische Universiteit Eindhoven
University of Technology

December 7th, 2010

Contents

Technische Universiteit Eindhoven University of Technology

Problem 1 - Illegal redistribution

Half-Life 2 code leaked online

The makers of the eagerly awaited Half-Life 2 have appealed for help to track down who leaked the source code of the game on the internet.

The game is eagerly awaited by fans

The software is not the full game but contains core information about it.

Valve, the makers of Half-Life 2, said the leak followed a concerted hacking effort on the company's computers over a number of months.

The new game had originally been due for release at the end of September before being knocked back to Christmas.

But the leak of the source code of Half-Life 2 has raised fears of a further postponement.

Developers Valve have confirmed that the software that has appeared on the net is indeed the computer code behind the game.

Huge Wikileaks release shows US 'ignored Iraq torture'

Wikileaks has released almost 400,000 secret US military logs, which suggest US commanders ignored evidence of torture by the Iraqi authorities.

The documents also suggest "hundreds" of civilians were killed at US military checkpoints after the invasion in 2003.

And the files show the US kept records of civilian deaths, despite previously denying it. The death toll was put at 109,000, of whom 66,081 were civilians.

Harry Potter film excerpt leaked online

A 36-minute clip of the latest Harry Potter movie has been leaked online ahead of its international release.

Harry Potter and the Deathly Hallows Part 1 is out in cinemas this week

Warner Bros said it was "working actively" to remove the video, which it said was "stolen and illegally posted" on file-sharing websites on Tuesday.

"We are vigorously investigating this matter and will prosecute those involved to the full extent of the law," it added in a statement.

Miljoenennota uitgelekt via RTL

Uitgegeven: 15 september 2007 19:59
Laatst gewijzigd: 15 september 2007 21:02

DEN HAAG - De miljoenennota is zaterdag uitgelekt via RTL Nieuws. Evenals in enkele voorgaande jaren slaagde de redactie van het programma erin het stuk te pakken te krijgen voor Prinsjesdag.

Sinds vrijdag beschikken de fractievoorzitters in de Eerste en de Tweede Kamer over een exemplaar onder embargo. De andere Kamerleden krijgen pas dinsdagochtend een embargo-exemplaar, evenals de pers.

Dinsdagmiddag presenteert minister Wouter Bos van Financiën het stuk in de Tweede Kamer. Vorige jaar lekte de miljoenennota niet uit. Toen werden er geen embargo-exemplaren verstrekt.

Solution 1 - Embed watermarks

Add unique fingerprints (watermarks) to each copy.

PvdA-Kamerlid Paul Tang bekent lekken begroting

Uitgegeven: 14 september 2009 16:20
Laatst gewijzigd: 15 september 2009 14:48

DEN HAAG - Tweede Kamerlid en financieel woordvoerder Paul Tang van de PvdA heeft bekend dat hij begrotingscijfers voor 2010 heeft gelekt aan RTL Nieuws.

© ANP

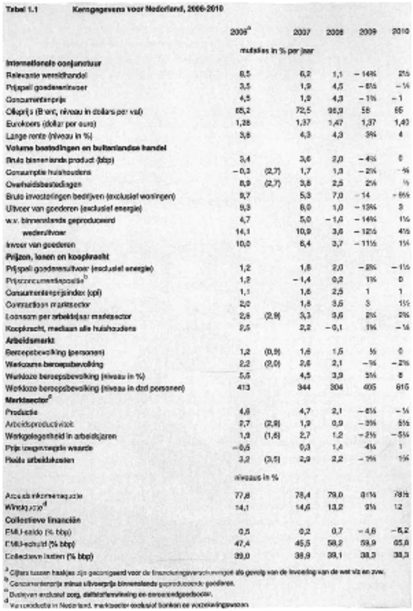
Tang heeft hiervoor maandag zijn excuses aangeboden aan Kamervoorzitter Gerdi Verbeet. Hij zal ook zijn verontschuldigingen aanbieden aan premier Jan Peter Balkenende.

Het fractiebestuur keurt het lekken af en straft Tang daarvoor. Het Kamerlid mag een maand lang niet het woord voeren op zijn beleidsterrein.

Geen Stijl

De website GeenStijl ontdekte bij uitvergroting van de gelekte stukken op de website van RTL Nieuws dat het watermerk 'PvdA' naar voren kwam.

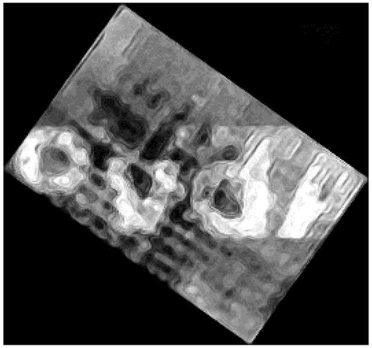
Daarop bekende Tang dat hij Macro-economische Verkenningen aan RTL heeft gegeven.



Tabel 1.1 Kerngegevens voor Nederland, 2006-2010

	2006[a]		2007	2008	2009	2010
	mutaties in % per jaar					
Internationale conjunctuur						
Relevante wereldhandel	8,5		6,2	1,1	− 14¾	2½
Prijspeil goedereninvoer	3,5		1,9	4,5	− 6½	− ¼
Consumentenprijs	4,5		1,9	4,3	− 1½	− 1
Olieprijs (Brent, niveau in dollars per vat)	65,2		72,5	98,9	58	65
Eurokoers (dollar per euro)	1,26		1,37	1,47	1,37	1,40
Lange rente (niveau in %)	3,8		4,3	4,3	3¾	4
Volume bestedingen en buitenlandse handel						
Bruto binnenlands product (bbp)	3,4		3,6	2,0	− 4¾	0
Consumptie huishoudens	− 0,3	(2,7)	1,7	1,3	− 2¾	− ¾
Overheidsbestedingen	8,9	(2,7)	3,8	2,5	2¾	½
Bruto investeringen bedrijven (exclusief woningen)	8,7		5,3	7,0	− 14	− 6½
Uitvoer van goederen (exclusief energie)	9,3		6,0	1,0	− 13¾	3
w.v. binnenlands geproduceerd	4,7		5,0	− 1,6	− 14¾	1½
wederuitvoer	14,1		10,9	3,6	− 12½	4½
Invoer van goederen	10,0		6,4	3,7	− 11½	1¼
Prijzen, lonen en koopkracht						
Prijspeil goederenuitvoer (exclusief energie)	1,2		1,6	2,0	− 2¾	− 1½
Prijsconcurrentiepositie[b]	1,2		− 1,4	0,2	1¾	0
Consumentenprijsindex (cpi)	1,1		1,6	2,5	1	1
Contractloon marktsector	2,0		1,6	3,5	3	1½
Loonsom per arbeidsjaar marktsector	2,6	(2,9)	3,3	3,6	2¾	2¾
Koopkracht, mediaan alle huishoudens	2,5		2,2	− 0,1	1¾	− ½
Arbeidsmarkt						
Beroepsbevolking (personen)	1,2	(0,9)	1,6	1,5	½	0
Werkzame beroepsbevolking	2,2	(2,0)	2,6	2,1	− ¾	− 2¾
Werkloze beroepsbevolking (niveau in %)	5,5		4,5	3,9	5¼	8
Werkloze beroepsbevolking (niveau in dzd personen)	413		344	304	405	615
Marktsector[c]						
Productie	4,6		4,7	2,1	− 6½	− ½
Arbeidsproductiviteit	2,7	(2,9)	1,9	0,9	− 3¾	5½
Werkgelegenheid in arbeidsjaren	1,9	(1,6)	2,7	1,2	− 2½	− 5½
Prijs toegevoegde waarde	− 0,5		0,3	1,4	4½	1
Reële arbeidskosten	3,2	(3,5)	2,9	2,2	− 1¾	1¾
	niveaus in %					
Arbeidsinkomensquote	77,8		78,4	79,0	81¼	78½
Winstquote[d]	14,1		14,6	13,2	9½	12
Collectieve financiën						
EMU-saldo (% bbp)	0,5		0,2	0,7	− 4,6	− 6,2
EMU-schuld (% bbp)	47,4		45,5	58,2	59,9	65,0
Collectieve lasten (% bbp)	39,0		38,9	39,1	38,3	38,3

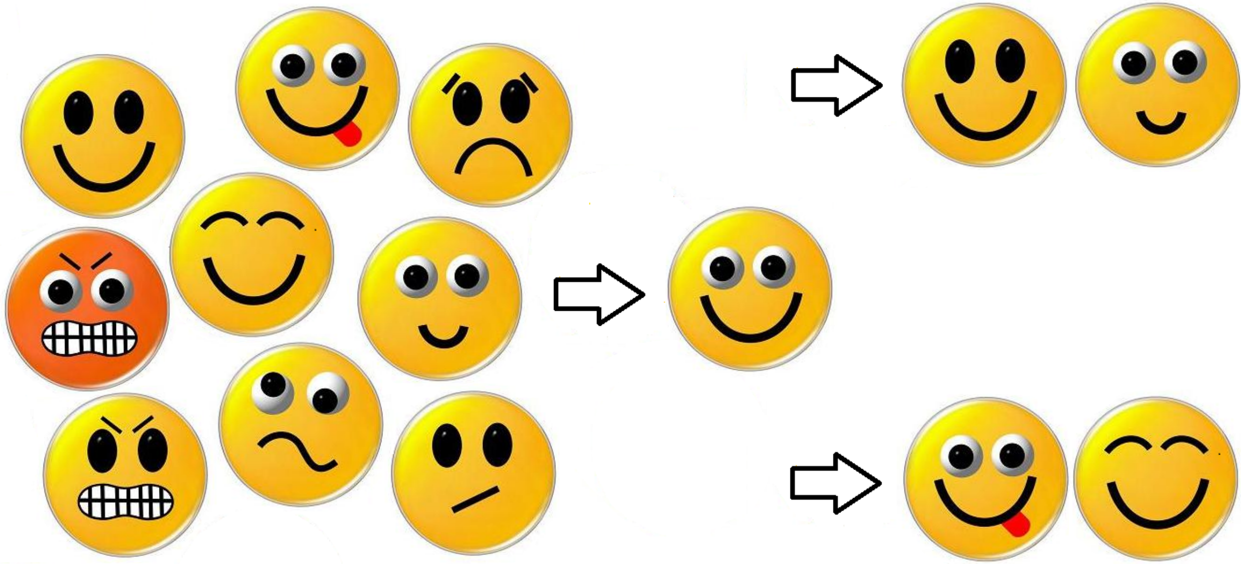
[a] Cijfers tussen haakjes zijn gecorrigeerd voor de financieringsverschuivingen als gevolg van de invoering van de wet zvz en zvw.
[b] Consumentenprijs minus uitvoerprijs binnenlands geproduceerde goederen.
[c] Bedrijven exclusief zorg, delfstoffenwinning en onroerendgoedsector.
[d] Van productie in Nederland, marktsector exclusief banken en verzekeringswezen.



This works only if it is hard to detect, edit and/or remove the watermarks.

Problem 2 - Collusion-attacks

Colluders compare their copies, searching for differences. Since their data is the same, the differences must be part of the watermark.



Colluders can then detect and edit that part of the fingerprint, making it hard to trace them.

Solution 2 - Fingerprinting codes

What makes the problem so hard?

- If the fingerprints are very different, then it is easy for colluders to detect and edit big parts of the watermark
- If the fingerprints are very similar, then it is hard to distinguish between users and get accurate accusations

But using smart mathematical techniques, we can construct fingerprinting schemes resistant against collusion attacks.

Abstraction

- Set of users $U = \{1, \ldots, n\}$
- Coalition or traitors $C = \{j_1, \ldots, j_c\} \subseteq U$
- Fingerprinting code $\mathcal{X}$: Codewords (vectors) over some alphabet Q
  - Alphabet size: q symbols ($|Q| = q$)
  - Codelength: ℓ positions ($\vec{x} \in Q^\ell$)
  - Cardinality: n users ($\mathcal{X} = \{\vec{x}_1, \ldots, \vec{x}_n\}$)
- Code $\mathcal{X}$ in matrix form: $X \in Q^{n \times \ell}$

$$X = \begin{pmatrix} \longleftarrow & \vec{x}_1 & \longrightarrow \\ & \vdots & \\ \longleftarrow & \vec{x}_n & \longrightarrow \end{pmatrix} \text{ e.g. } X = \begin{pmatrix} 0 & 2 & 1 & 1 & 2 & 1 & 3 \\ 3 & 1 & 2 & 0 & 0 & 0 & 2 \\ 3 & 3 & 2 & 0 & 1 & 0 & 1 \\ 2 & 3 & 1 & 2 & 2 & 2 & 1 \end{pmatrix} \in \{0, \ldots, 3\}^{4 \times 7}$$

- Coalition generates forgery $\vec{y}$ using some pirate strategy ρ

Pirate strategy

Assumptions on what pirates can do:

- If a coalition sees symbols $S \subseteq Q$ on position i ($|S| > 1$), then...
  - Restricted digit model: $\vec{y}_i \in S$.
  - Arbitrary digit model: $\vec{y}_i \in Q$.
  - Allowing erasures: $\vec{y}_i \in S \cup \{?\}$ (or $\vec{y}_i \in Q \cup \{?\}$)
  - Binary alphabet: All equivalent
- Marking Assumption: If $S = \{\sigma\}$ then $\vec{y}_i = \sigma$
- Secret embedding of fingerprints in data is perfect (not our problem)

Example:

$$\begin{aligned} X(C) &= \begin{pmatrix} 0 & 2 & \mathbf{1} & 1 & \mathbf{2} & 1 & 3 \\ 2 & 3 & \mathbf{1} & 2 & \mathbf{2} & 2 & 1 \end{pmatrix} \\ \vec{y} &= \begin{pmatrix} 0 & 3 & \mathbf{1} & 2 & \mathbf{2} & 1 & 3 \end{pmatrix} \end{aligned}$$

Some pirate strategies

Besides these assumptions, pirates can do anything they want. Suppose $q = 2$ and $0 < k < c$ is the number of ones seen by C ($k = 0$ or $k = c$: Marking Assumption).

- Random: $y_i \in_R \{0, 1\}$
- Always 1: $y_i = 1$
- Majority: $y_i = 1$ if $k > c/2$ and $y_i = 0$ if $k < c/2$
  - Majority/one: If $k = c/2$, $y_i = 1$
  - Majority/first: If $k = c/2$, $y_i = \sigma_1$
  - Majority/random: If $k = c/2$, $y_i \in_R \{0, 1\}$
- Minority: $y_i = 1$ if $k < c/2$ and $y_i = 0$ if $k > c/2$
  - Minority/...
- Interleaving: $\mathbb{P}[y_i = 1] = k/c$ (i.e. $y_i \in_R \{\sigma_1, \ldots, \sigma_c\}$)
- Scapegoat: $y_i = \sigma_1$

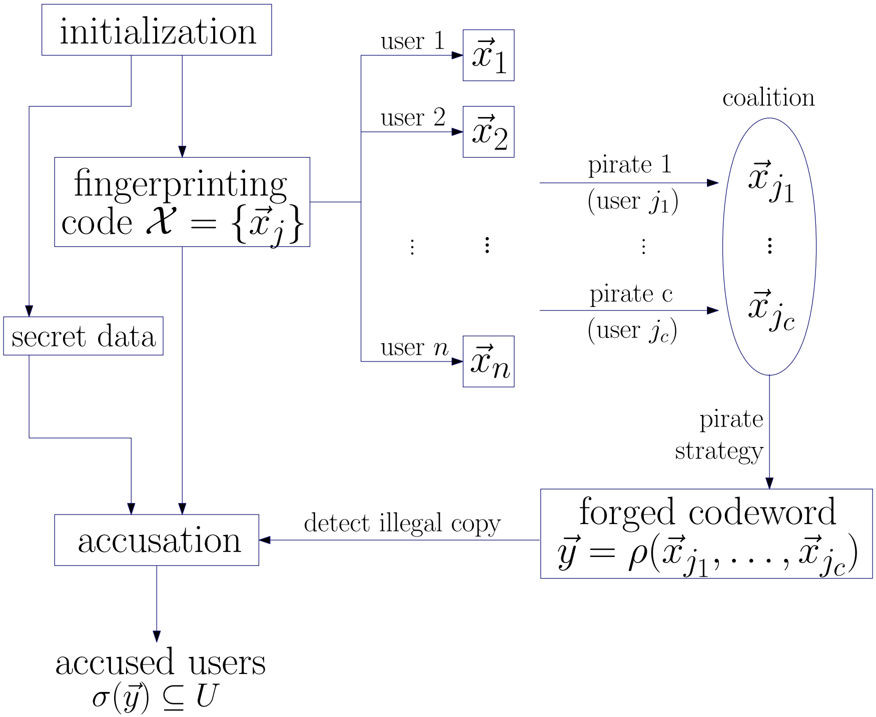
The scheme should be secure against all attacks.

What do we want?

- Resistancy against many colluders
- Resistancy against any pirate strategy
- Short codelength
  - Less redundant data
- Small alphabet
  - In practice: Bandwidth needed linear in alphabet size
- Avoid accusing innocent users
- Accuse at least one guilty user (preferably more)

Static schemes

- Send codewords
- Coalition produces some forgery
- Receive forgery
- Accuse certain users
- Advantages:
  - Many applications
  - Only one codeword
- Disadvantages:
  - More data needed
  - Catch few colluders



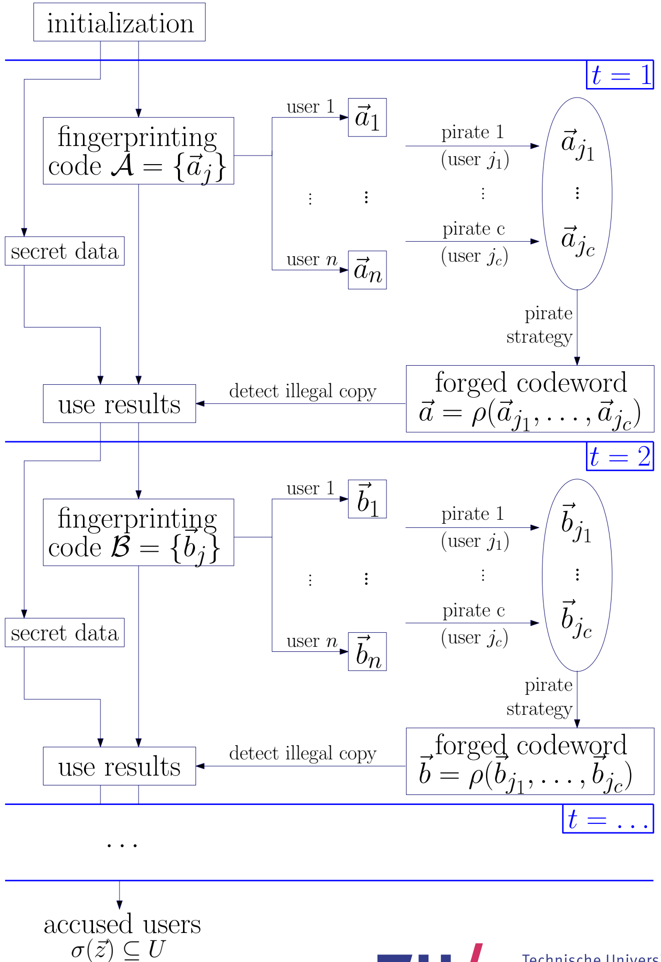
Technische Universiteit Eindhoven University of Technology

Dynamic schemes

- Send first set of codewords
- Receive first forgery
- Send new codewords

  ⋮

- Receive final forgery
- Accuse users based on all results
- Advantages:
  - Less total data needed
  - Possibly catch all colluders
- Disadvantages:
  - Only few applications
  - Computations required during the broadcast



Deterministic/probabilistic schemes

Deterministic schemes: No error

- Always absolute certainty
- Soundness: Never accuse any innocent users
- Completeness: Always accuse at least one guilty user
- Always alphabet size $q \geq c + 1$
- Works only in restricted digit model

Probabilistic schemes: Error bounded by $\epsilon > 0$

- Small probability of error
- Soundness: Accuse no innocent users with probability at least $1 - \epsilon$
- Completeness: Accuse a guilty user with probability at least $1 - \epsilon$
- Decoupling ϵ to ϵ_1 (Soundness) and ϵ_2 (Completeness): $\epsilon_1 \ll \epsilon_2$
- Alphabet size $q \geq 2$
- Works against any attack model

Deterministic static schemes (IPP codes)

- Identifiable Parent Property: Always identify a "parent"
- Advantages:
  - No error, always absolute certainty
  - Only one codeword necessary
- Disadvantages:
  - Large alphabet size ($q \geq c+1$, or even $q \geq c^2$)
  - Long codelength
- Lower bounds on codelength:
  - $\ell = \Omega(c \log(n/c)/\log(q))$ [Bla03b]
  - $\ell = \Omega(c^2 \log(n)/\log(q))$ [AS04]
- Upper bounds on codelength: (constructions)
  - $\ell = \mathcal{O}(c^2 \log(n)/\log(q))$ for $q = \mathcal{O}(c^2 \log(n))$ [SSW01]
  - $\ell = \mathcal{O}(c^2 \log(n)/\log(q \cdot g(c)))$ for any $q \geq c$, for some $g(c)$ [AS04]

IPP codes: Example

Tetracode; Hamming code [HVLLT98] [BEN07]
Only non-trivial "beautiful" code [BEN07]

- $n = 9$ users
- $c = 2$ colluders
- $q = 3$ alphabet size
- $\ell = 4$ codelength

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

Why is it secure against 2 colluders?

- Every two codewords have distance 3
- Every word has distance ≤ 1 to exactly one codeword

$$\begin{array}{rcl} \vec{a} & = & (\mathbf{1}, 0, \mathbf{1}, 2) \\ \vec{b} & = & (2, \mathbf{2}, \mathbf{1}, \mathbf{0}) \\ \hline \vec{y} & = & (1, 2, 1, 0) \\ & \rightarrow & d(\vec{y}, \vec{b}) = 1 \end{array}$$

IPP codes: Error-correcting codes

- If $\mathcal{C} = (\ell, K, d)_q$ is an error-correcting code of cardinality $n = K$ satisfying $d > \ell(1 - 1/c^2)$, then $\mathcal{C}$ is a c-IPP-code. [SSW01] [SNW03]
- If $q \geq \ell - 1$ and $k = \lceil \ell/c^2 \rceil$, then there exists a linear Reed-Solomon error-correcting code with parameters $[\ell, k, d]_q$ satisfying $d > \ell(1 - 1/c^2)$ of cardinality $n = q^k = q^{\lceil \ell/c^2 \rceil}$. [SSW01]
- If $q \geq \ell - 1$, then there exist c-IPP codes satisfying $n = q^{\lceil \ell/c^2 \rceil}$, i.e. $\ell = \mathcal{O}(c^2 \log(n))$ and $q = \mathcal{O}(c^2 \log(n))$.

Probabilistic static schemes

- Probabilistic static schemes: Static schemes with $\epsilon > 0$ error
- Advantages:
  - Small alphabet size ($q \geq 2$)
  - Short codelength
- Disadvantages:
  - Small probability of error ϵ
- Lower bounds on codelength:
  - $\ell = \Omega(c \log(1/c\epsilon))$ for $q = 2$ [BS98]
  - $\ell = \Omega(c^2 \log(1/\epsilon))$ for $q = 2$ [Tar03]
  - $\ell \geq 1.38c^2 \log(1/\epsilon)$ for $q = 2$ [HM09b]
- Upper bounds on codelength: (constructions)
  - $\ell = \mathcal{O}(c^4 \log(n/\epsilon) \log(1/\epsilon))$ for $q = 2$ [BS98]
  - $\ell = 100c^2 \log(1/\epsilon)$ for $q = 2$ [Tar03]
  - $\ell \approx 4.93c^2 \log(1/\epsilon)$ for $q = 2$ and $c \to \infty$ [SKC08]

Technische Universiteit Eindhoven University of Technology

1. Initialization: Choose the codelength $\ell(c, \epsilon)$ and parameters $t(c), Z(c, \epsilon)$, and choose probabilities $p_i \sim F_t$.
2. Codeword generation: Choose the symbols X_{ji} by $X_{ji} \sim \text{Ber}(p_i)$.

	Position 1	Position 2	$\ldots$	Position ℓ
Probability p_i	$p_1 \sim F_t$	$p_2 \sim F_t$	$\ldots$	$p_\ell \sim F_t$
User 1	$X_{1,1} \sim \text{Ber}(p_1)$	$X_{1,2} \sim \text{Ber}(p_2)$	$\ldots$	$X_{1,\ell} \sim \text{Ber}(p_\ell)$
User 2	$X_{2,1} \sim \text{Ber}(p_1)$	$X_{2,2} \sim \text{Ber}(p_2)$	$\ldots$	$X_{2,\ell} \sim \text{Ber}(p_\ell)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
User n	$X_{n,1} \sim \text{Ber}(p_1)$	$X_{n,2} \sim \text{Ber}(p_2)$	$\ldots$	$X_{n,\ell} \sim \text{Ber}(p_\ell)$

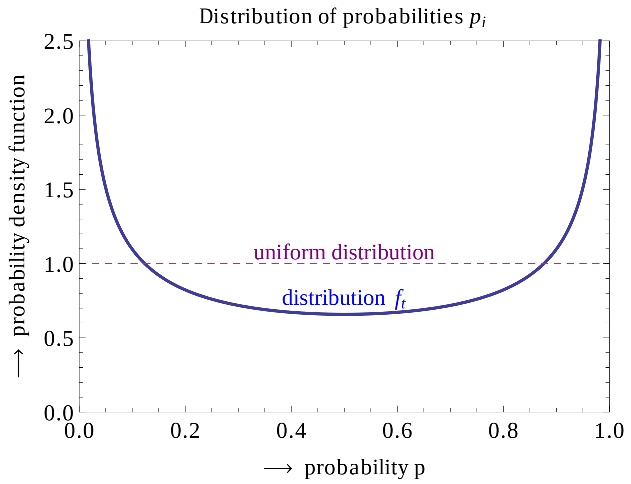
The Tardos scheme - Rough outline

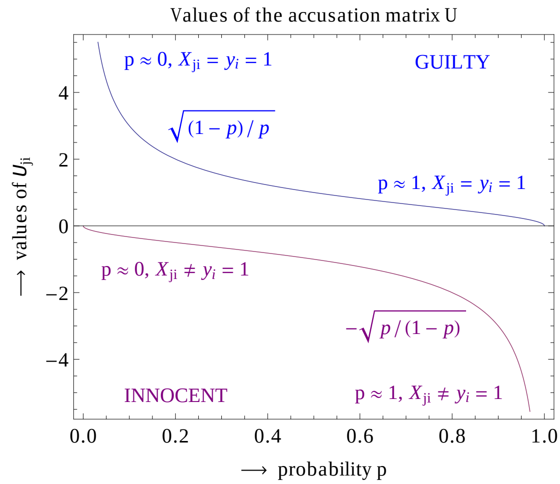
1. Initialization: Choose the codelength $\ell(c, \epsilon)$ and parameters $t(c), Z(c, \epsilon)$, and choose probabilities $p_i \sim F_t$.
2. Codeword generation: Choose the symbols X_{ji} by $X_{ji} \sim \text{Ber}(p_i)$.

	Position 1	Position 2	...	Position ℓ
Probability p_i	$p_1 = 0.03$	$p_2 = 0.81$	...	$p_\ell = 0.1$
User 1	$X_{1,1} = 0$	$X_{1,2} = 1$	...	$X_{1,\ell} = 0$
User 2	$X_{2,1} = 0$	$X_{2,2} = 0$	...	$X_{2,\ell} = 0$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
User n	$X_{n,1} = 0$	$X_{n,2} = 1$	...	$X_{n,\ell} = 1$

The Tardos scheme - Rough outline

1. Initialization: Choose the codelength $\ell(c,\epsilon)$ and parameters $t(c), Z(c,\epsilon)$, and choose probabilities $p_i \sim F_t$.
2. Codeword generation: Choose the symbols X_{ji} by $X_{ji} \sim \mathrm{Ber}(p_i)$.
3. Accusation (precomputation): Calculate the accusation matrix U by $U_{ji} = +\sqrt{(1-p)/p}$ if $X_{ji} = 1$ and $U_{ji} = -\sqrt{p/(1-p)}$ if $X_{ji} = 0$
4. Accusation (given forgery $\vec{y}$): Accuse user j if $S_j = (U\vec{y})_j > Z$.





The Tardos scheme - Dummy example

Dummy parameters: $n = 5, \ell = 6, Z = 1, \vec{p} = (0.8, 0.7, 0.2, 0.1, 0.5, 0.3)$

$$X = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}, \quad U \approx \begin{pmatrix} 0.5 & 0.7 & -0.5 & -0.3 & -1.0 & 1.5 \\ -2.0 & 0.7 & -0.5 & -0.3 & 1.0 & -0.7 \\ 0.5 & -1.5 & -0.5 & -0.3 & 1.0 & 1.5 \\ 0.5 & -1.5 & 2.0 & -0.3 & -1.0 & -0.7 \\ 0.5 & 0.7 & -0.5 & -0.3 & 1.0 & -0.7 \end{pmatrix}$$

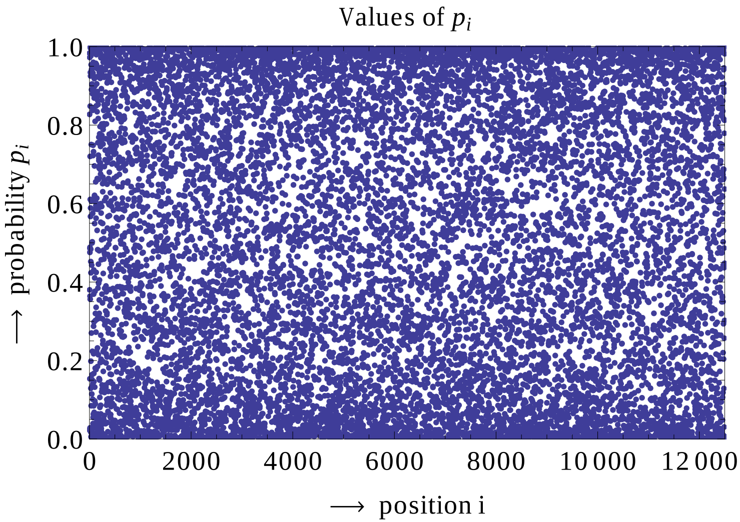
Some examples of forgeries and accusations:

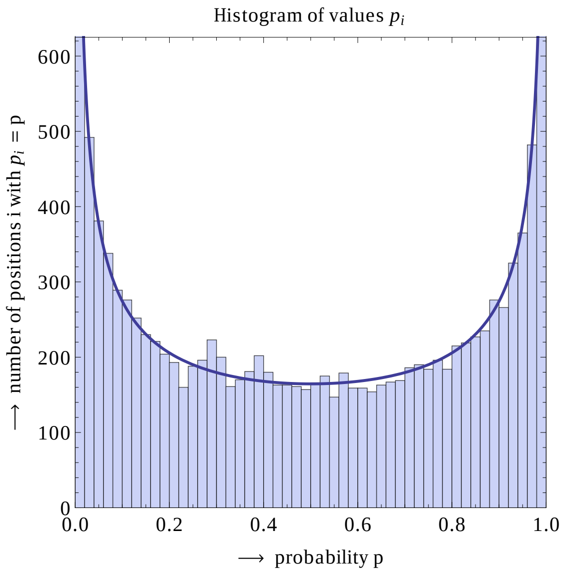
Forgery $\vec{y}$	S_1	S_2	S_3	S_4	S_5	$\sigma(\vec{y})$	Comment
$(0,1,1,0,0,1)$	**1.7**	−0.5	−0.5	−0.2	−0.5	$\{1\}$	
$(0,1,0,0,1,0)$	−0.3	**1.7**	−0.5	−2.5	**1.7**	$\{2,5\}$	
$(1,0,0,1,1,0)$	−0.8	−1.3	**1.2**	−0.8	**1.2**	$\{3,5\}$	impossible!
$(0,0,0,0,0,0)$	0	0	0	0	0	$\emptyset$	always no one
$(1,1,1,1,1,1)$	0.9	−1.8	0.7	−1.0	0.7	$\emptyset$	
$(1,1,1,0,1,1)$	**1.2**	−1.5	**1.0**	−0.7	**1.0**	$\{1,3,5\}$	removed a 1
$(1,1,1,0,0,1)$	**2.2**	−2.5	0.0	0.3	0.0	$\{1\}$	removed a 1
$(0,0,1,0,0,0)$	−0.5	−0.5	−0.5	**2.0**	−0.5	$\{4\}$	user 4 accused

The Tardos scheme - Real example

Scheme parameters: $\epsilon = e^{-5} \approx 0.0067, c = 5, n = 100 \Rightarrow \ell = 12500, t = 1/1500, Z = 500, p_i \sim F_t, X \in \{0,1\}^{100\times 12500}, U \in \mathbb{R}^{100\times 12500}$ (then U already contains 1.250.000 real numbers)

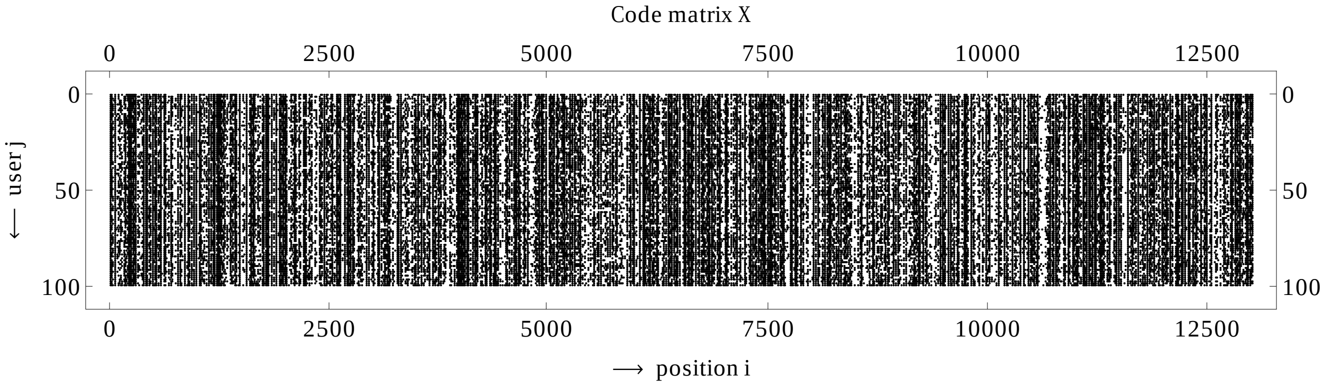
Simulations (interleaving attack): Select an arbitrary coalition, calculate $\vec{y}$, calculate $\sigma(\vec{y})$ and see if the accusation worked

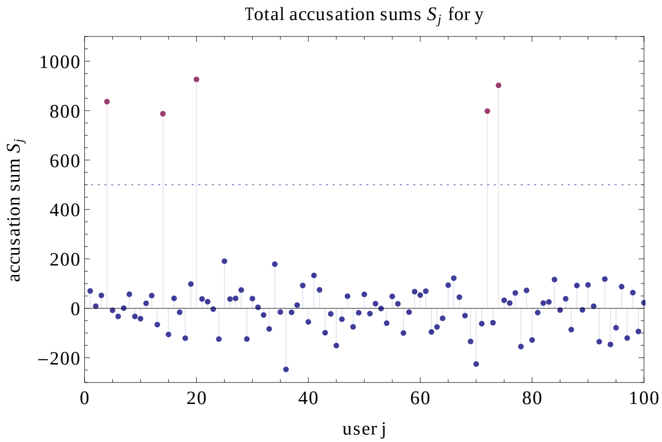


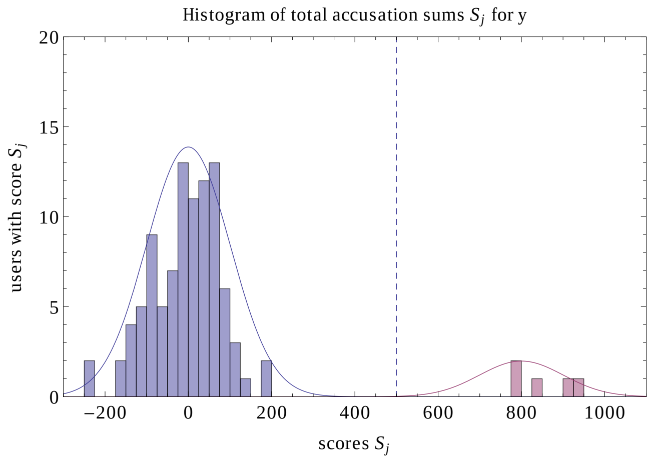


TU/e Technische Universiteit Eindhoven University of Technology

The Tardos scheme - Real example



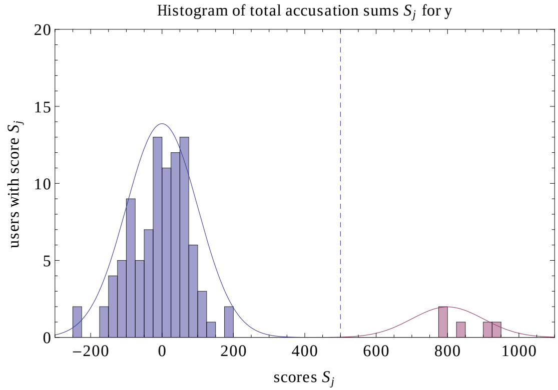




Technische Universiteit Eindhoven University of Technology

The Tardos scheme - Why does it work?

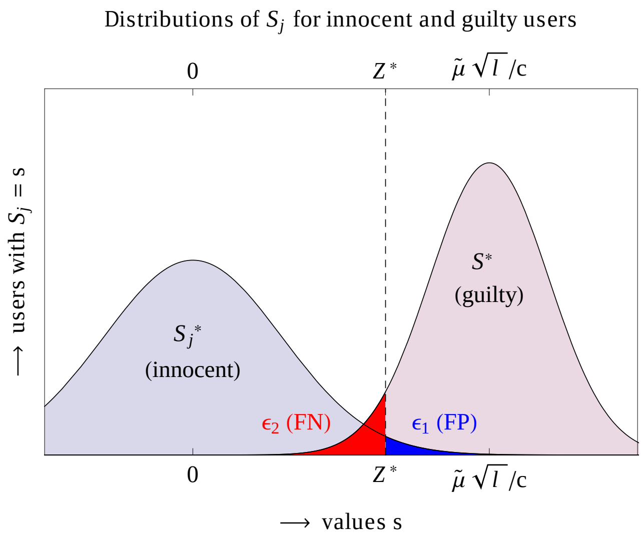
- Why are no innocent users accused?
  - All codewords are independent, so it is impossible to frame anyone
  - Positive and negative contributions outweigh eachother
  - S_j is roughly distributed as $\mathcal{N}(0, \sqrt{\ell})$ while $Z \gg \sqrt{\ell}$
- Why are guilty users accused?
  - On detectable positions, pirates cannot decrease $S = \sum_{j \in C} S_j$
  - On undetectable positions, S definitely increases
  - $\frac{S}{c}$ is roughly distributed as $\mathcal{N}(\tilde{\mu}\ell/c, \tilde{\sigma}^2\ell/c^2)$ while $Z \ll \tilde{\mu}\ell/c$

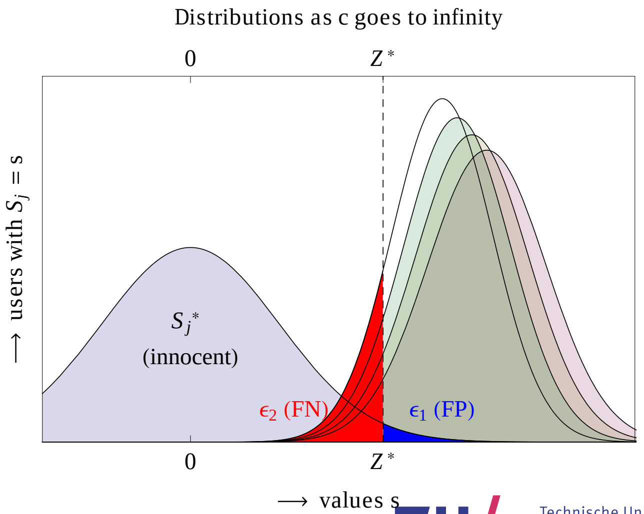


The Tardos scheme - $\ell = \Theta(c^2)$, $Z = \Theta(c)$

Suppose $S_j^* = S_j/\sqrt{\ell} \sim \mathcal{N}(0,1)$, $S^* = \frac{S}{c}/\sqrt{\ell} \sim \mathcal{N}(\tilde{\mu}\sqrt{\ell}/c, \tilde{\sigma}^2/c^2)$, $Z^* = Z/\sqrt{\ell}$. Then we need $\ell = \Theta(c^2)$ and $Z = \Theta(c)$ for the scheme to work.

- If $\ell = o(c^2)$ then $\mathbb{E}[S^*] \to 0$ and $\mathrm{Var}[S^*] \to 0$ as $c \to \infty$
- If $\ell = \Theta(c^2)$ then $\mathbb{E}[S^*] \to L$ and $\mathrm{Var}[S^*] \to 0$ as $c \to \infty$
- If $Z = o(\sqrt{\ell})$ then $Z^* \to 0 = \mathbb{E}[S_j^*]$ so $\epsilon_1 \to 1/2$ which is bad
- If $Z > \Omega(\sqrt{\ell})$ then $Z^* \to \infty > \mathbb{E}[S^*]$ so $\epsilon_2 \to 1$ which is bad





The Tardos scheme - Improvements

Suggested improvements:

- Use symmetric accusation function instead of U [SVCT06]
- Tighten the analysis in the proof [SVCT06], [BT08]
- Use the Gaussian approximation to estimate error probabilities [SS10]
- Use a discrete optimal distribution F_t [NFH+09]

With these optimizations, the factor 100 has been reduced to less than 5 in the asymptotic case of $c \to \infty$

Intermezzo: Irdeto's scheme

Irdeto's implementation: Uniformly random bits, accusation weights Hamming distance between the forgery and the codeword (simply count the number of matches), accuse if these weights are too large. This is a special case of Tardos' scheme with $F \equiv 1/2$. But is it safe?

Minority attack, 3 traitors:

Received symbols	Output	Matches	Differences	Increase in S
$0,0,0$	0	3	0	$+3$
$0,0,1$	1	1	2	-1
$0,1,0$	1	1	2	-1
$0,1,1$	0	1	2	-1
$1,0,0$	1	1	2	-1
$1,0,1$	0	1	2	-1
$1,1,0$	0	1	2	-1
$1,1,1$	1	3	0	$+3$
		12	12	Total: 0

Intermezzo: Irdeto's scheme

Irdeto's implementation: Uniformly random bits, accusation weights Hamming distance between the forgery and the codeword (simply count the number of matches), accuse if these weights are too large. This is a special case of Tardos' scheme with $F \equiv 1/2$. But is it safe?

Minority attack, 5 traitors:

Received symbols	Output	Matches	Differences	Increase in S
$0,0,0,0,0$	0	5	0	$+5$
$0,0,0,0,1$	1	1	4	-3
$0,0,0,1,0$	1	1	4	-3
$0,0,0,1,1$	1	2	3	-1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$1,1,1,1,0$	0	1	4	-3
$1,1,1,1,1$	0	5	0	$+5$
		70	90	Total: -20

Deterministic dynamic schemes

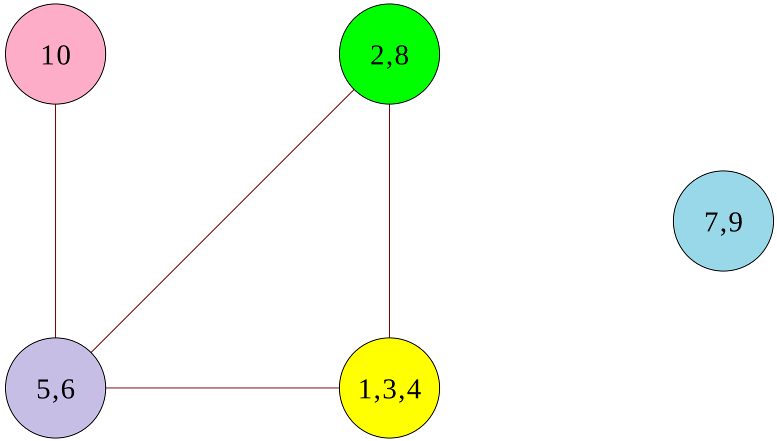
- Restricted digit model: only symbols of coalition allowed
- Advantages:
  - No error
  - Shorter time than length in static schemes
  - Catch all colluders with same effort
  - Works against any number of colluders; c need not be known
- Disadvantages:
  - Only works dynamically
  - Large alphabet size ($q > c$)
- Upper bounds on codelength, time: (constructions)
  - $q = 2c + 1, t \leq c \log(n) + c$ [FT01]
  - $q = c + 1, t = \mathcal{O}(c^3 \log(n))$ [BPS00]
  - $q = c + 1, t = \mathcal{O}(c^2 + c \log(n))$ [BPS00]

Technische Universiteit Eindhoven University of Technology

General graph notation

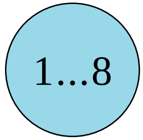
Graph description: Vertices (points) V, edges (lines) E

- Vertices: Disjoint subsets of U (forms a partition of U)
- Edges: If $S \sim T$ then $S \cup T$ contains at least one pirate
- Vertex colors: Colors correspond to symbols
- A vertex S gets color c if all users in S get symbol c
- At least c pirates $\Leftrightarrow$ Any vertex cover has size at least c



The Fiat-Tassa scheme

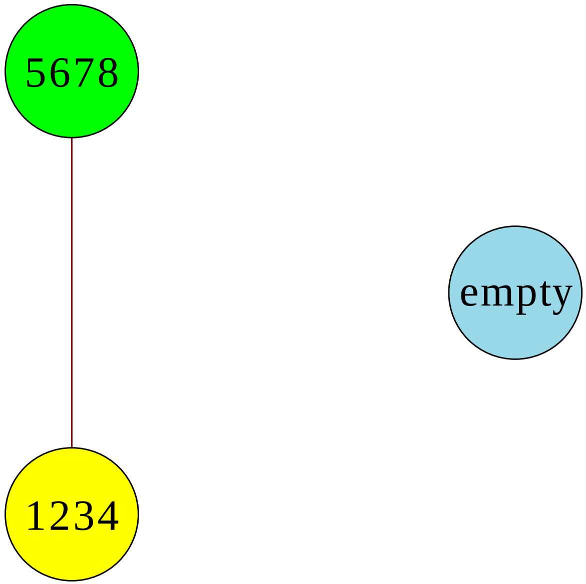
Example: 8 users, 2 traitors (users 2 and 5)
Colors seen by coalition: Blue, Blue



Output color: Blue

The Fiat-Tassa scheme

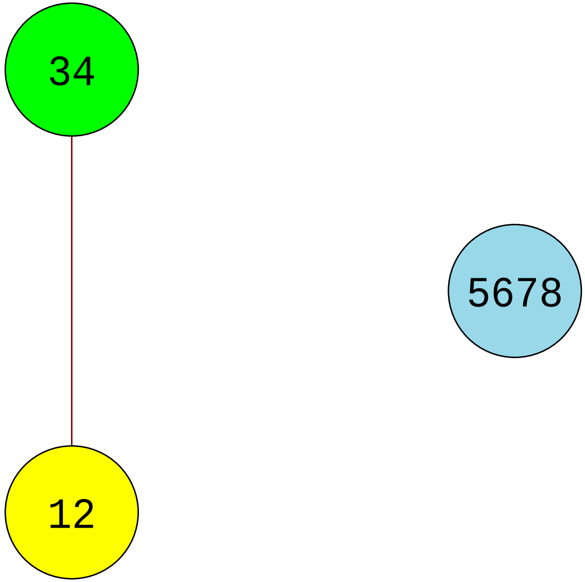
Example: 8 users, 2 traitors (users 2 and 5)
Colors seen by coalition: Yellow, Green



Output color: Yellow

The Fiat-Tassa scheme

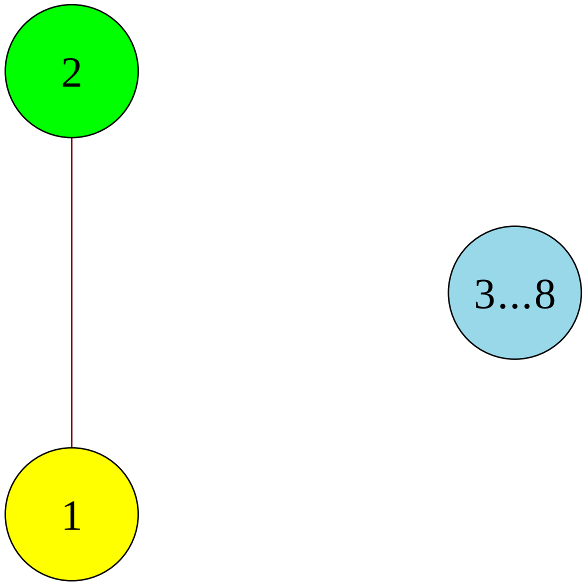
Example: 8 users, 2 traitors (users 2 and 5)
Colors seen by coalition: Yellow, Blue



Output color: Yellow

The Fiat-Tassa scheme

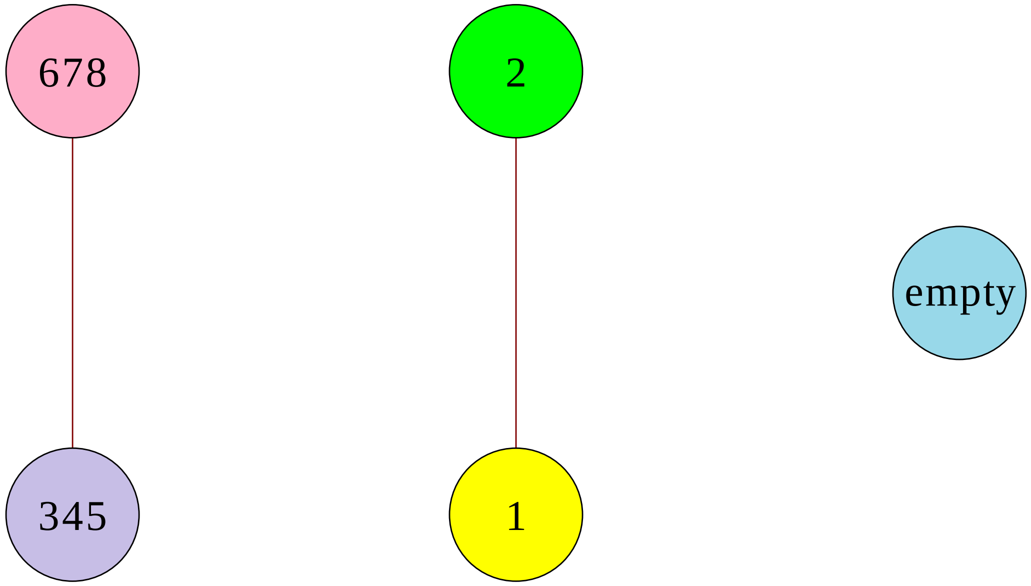
Example: 8 users, 2 traitors (users 2 and 5)
Colors seen by coalition: Green, Blue



Output color: Blue

The Fiat-Tassa scheme

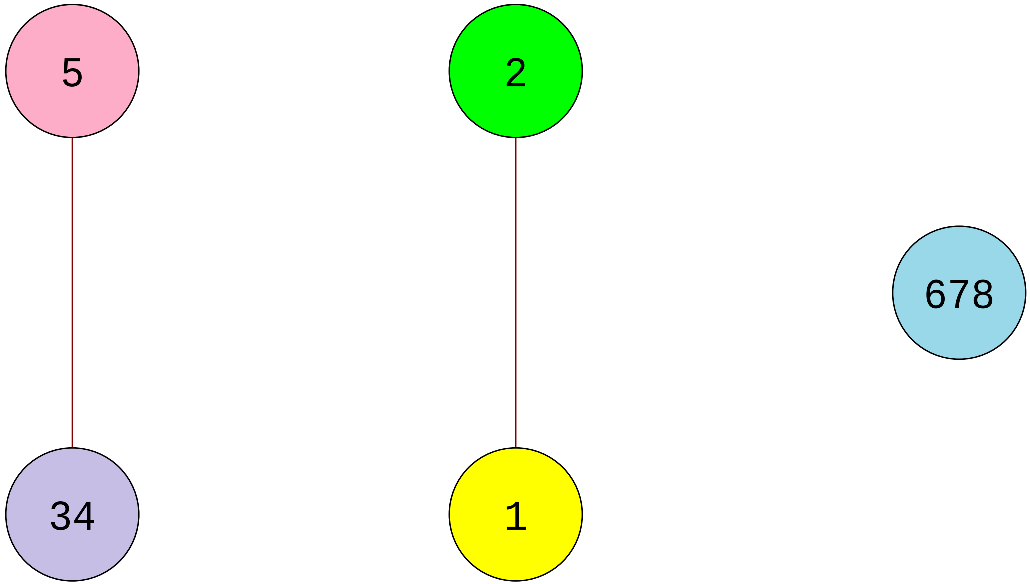
Example: 8 users, 2 traitors (users 2 and 5)
Colors seen by coalition: Green, Purple



Output color: Purple

The Fiat-Tassa scheme

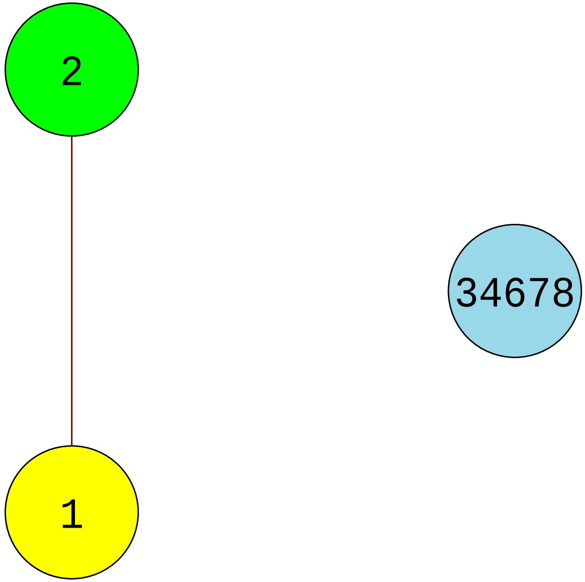
Example: 8 users, 2 traitors (users 2 and 5)
Colors seen by coalition: Green, Rose



Output color: Rose

The Fiat-Tassa scheme

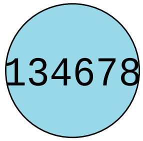
Example: 8 users, 2 traitors (users 2 and 5)
Colors seen by coalition: Green



Output color: Green

The Fiat-Tassa scheme

Example: 8 users, 2 traitors (users 2 and 5)
Colors seen by coalition: None

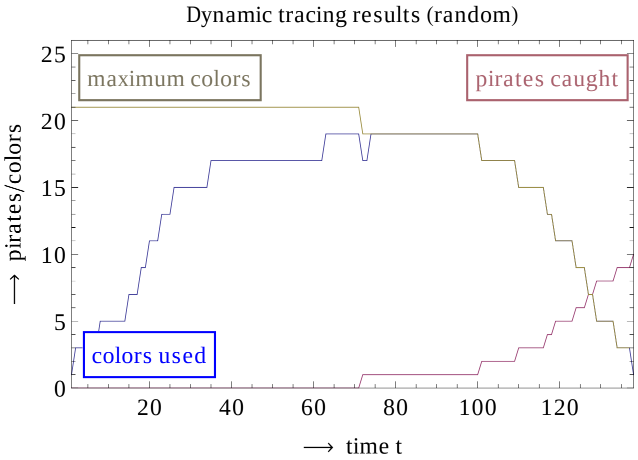


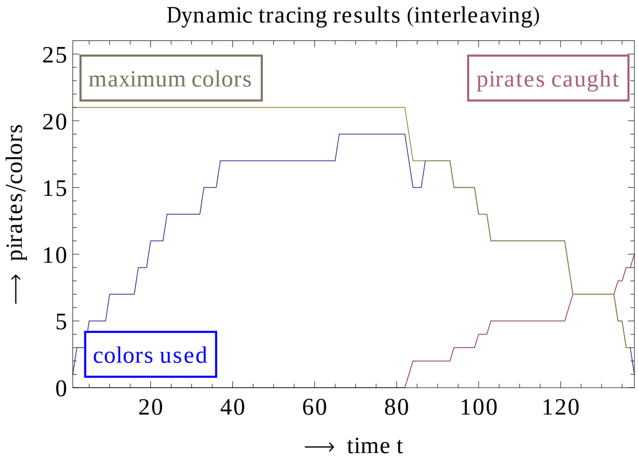
Output color: None

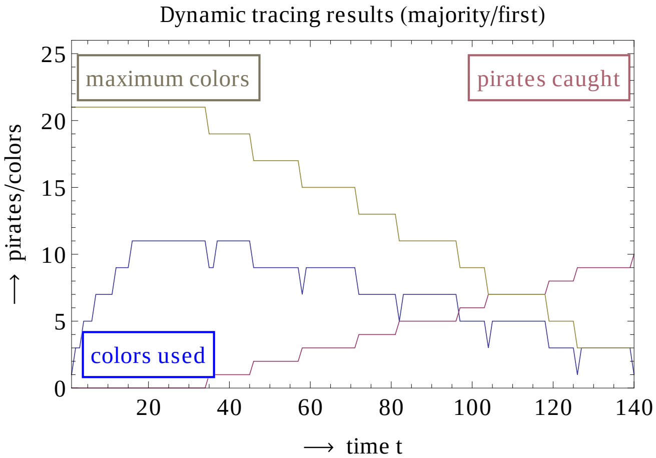
The Fiat-Tassa scheme

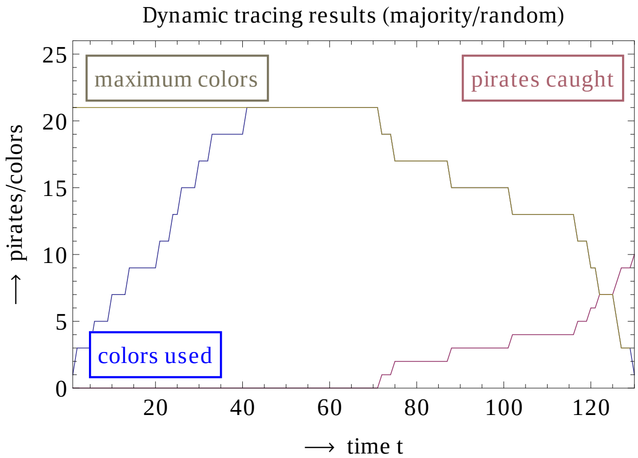
- Isolating a single traitor: At most $t = \log_2(n)$ steps
- Tracing at least one traitor: At most $t = \log_2(n) + 1$ steps
- Tracing all traitors: At most $t = c\log_2(n) + c$ steps
- Colors needed (alphabet size): At most $q = 2c + 1$ (2 for each traitor, 1 for not yet suspected users)
- Using certain pirate strategies, these bounds are also "often" achieved

The Fiat-Tassa scheme - Simulations



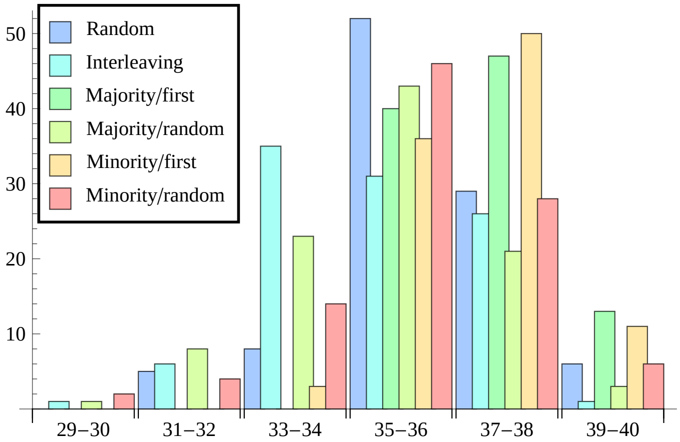


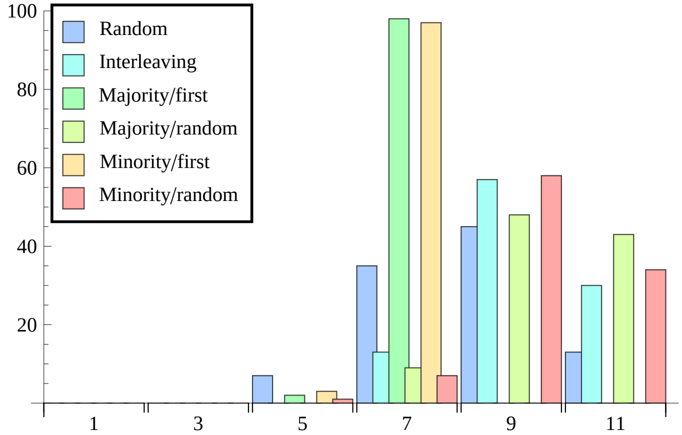




The Fiat-Tassa scheme - Strategies

Compare strategies for $n = 100, c = 5$ with 100 simulations.
Left: Time used for different pirate strategies.
Right: Maximum number of colors used during a tracing process.





The Berkman-Parnas-Sgall scheme

Certain pairs of not yet connected vertices get the same color. If a received color belongs to two vertices, add an edge between the two vertices.

- Degree algorithm: $t = \mathcal{O}(c^3 \log(n)), q = c + 1$
  - Keep adding edges until vertices get high enough degrees
  - If a vertex has degree $d > c$, then it must be guilty
  - Enough pairs of unconnected vertices always exist
  - Complication: c may not be known
- Clique algorithm: $t = \mathcal{O}(c^3 \log(n)), q = c + 1$
  - Keep adding edges until cliques occur (clique: complete subgraph)
  - Any clique of size k contains at least $k - 1$ traitors
  - At some point, traitors will be alone in a set and caught
- Optimal algorithm: $t = \mathcal{O}(c^2 + c \log(n)), q = c + 1$
  - Very complicated extension of the clique algorithm

Probabilistic dynamic schemes

- Inner code of Fiat-Tassa scheme: IPP-code
  - $\mathcal{X} = \{0, 1, \ldots, q-1\}$
  - Constant codelength $\ell = 1$
  - Maximum alphabet size $q \leq 2c + 1$
- Replace with new inner code of hybrid scheme: Probabilistic code
  - $\mathcal{X} = \{\vec{x}_1, \ldots, \vec{x}_q\}$
  - Maximum codelength $\ell > 1$
  - Constant alphabet size $q \geq 2$
- Advantage: Small alphabet size ($q \geq 2$)
- Disadvantages:
  - Errors stack up, so it's hard to bound the error probability
  - Longer codelength and time needed
- Upper bound on codelength, time: (constructions)
  - $q = 2, t \cdot \ell = \mathcal{O}(c^4 \log(1/\epsilon))$ [Tas05]

The Tassa scheme

- Inner code of Tassa's hybrid scheme: Boneh-Shaw code
  - Maximum codelength $\ell = \mathcal{O}(c^3 \log(1/\epsilon))$
  - Constant alphabet size $q = 2$
- Total "effort" bounded by $t \cdot \ell = \mathcal{O}(c^4 \log^2(1/\epsilon))$
- Tassa's analysis also gives no better bound than $\mathcal{O}(c^4 \log^2(1/\epsilon))$

Summary

Let $\epsilon = \Theta(1/n)$ and $k = \log(1/\epsilon) = \Theta(\log(n))$. Then:

	q	ϵ	ℓ	t	$\ell \cdot t$
Det. static	$\Omega(c)$	0	$\Omega(c^2k)$	$\mathcal{O}(1)$	$\Omega(c^2k)$
- [SSW01]	$\mathcal{O}(c^2k)$	0	$\mathcal{O}(c^2k)$	1	$\mathcal{O}(c^2k)$
- [AS04]	$\mathcal{O}(c)$	0	$\mathcal{O}(c^2k)$	1	$\mathcal{O}(c^2k)$
Prob. static	$\mathcal{O}(1)$	$\Omega(\epsilon)$	$\Omega(c^2k)$	$\mathcal{O}(1)$	$\Omega(c^2k)$
- [BS98]	2	$\mathcal{O}(\epsilon)$	$\mathcal{O}(n^3k^2)$	1	$\mathcal{O}(n^3k^2)$
- [BS98]	2	$\mathcal{O}(\epsilon)$	$\mathcal{O}(c^4k^2)$	1	$\mathcal{O}(c^4k^2)$
- [Tar03]	2	$\mathcal{O}(\epsilon)$	$\mathcal{O}(c^2k)$	1	$\mathcal{O}(c^2k)$
Det. dynamic	$\Omega(c+\alpha)$	0	$\mathcal{O}(1)$	$\Omega(c^2/\alpha + ck)$	$\Omega(c^2/\alpha + ck)$
- [FT01]	$2c+1$	0	1	$\mathcal{O}(ck)$	$\mathcal{O}(ck)$
- [BPS00]	$c+1$	0	1	$\mathcal{O}(c^3k)$	$\mathcal{O}(c^3k)$
- [BPS00]	$c+1$	0	1	$\mathcal{O}(c^2+ck)$	$\mathcal{O}(c^2+ck)$
Prob. dynamic	$\mathcal{O}(1)$	$\Omega(\epsilon)$	?	?	?
- [Tas05]	2	$\mathcal{O}(\epsilon)$	$\mathcal{O}(c^3k^2)$	$\mathcal{O}(ck)$	$\mathcal{O}(c^4k^3)$

Technische Universiteit Eindhoven University of Technology

Future work

- Investigate other options for a hybrid scheme
- Look at some more important papers
- Investigate practical implementation issues
- Consider the special (practical) case for $c = 5 \ldots 25$
- Run simulations with real values used in practice

Questions

Thank you for your attention!

Any questions?

References

[AB08] Prasanth Anthapadmanabhan and Alexander Barg. Randomized Frameproof Codes: Fingerprinting Plus Validation Minus Tracing. CoRR, abs/0802.3419, 2008.

[ABD06] Prasanth Anthapadmanabhan, Alexander Barg, and Ilya Dumer. On the Fingerprinting Capacity Under the Marking Assumption. CoRR, abs/cs/0612073, 2006.

[AFS01] Noga Alon, Eldar Fischer, and Mario Szegedy. Parent-Identifying Codes. Journal of Combinatorial Theory, Series A, 95(2):349–359, 2001.

[AS04] Noga Alon and Uri Stav. New Bounds on Parent-Identifying Codes: The Case of Multiple Parents. Comb. Probab. Comput., 13:795–807, 2004.

[AT09] Ehsan Amiri and Gabor Tardos. High rate fingerprinting codes and the fingerprinting capacity. Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, pages 336–345, 2009.

[BBK03] Alexander Barg, George Blakley, and Grigory Kabatiansky. Digital fingerprinting codes: Problem statements, constructions, identification of traitors. IEEE Trans. Inform. Theory, 49:852–865, 2003.

[BCE+01] Alexander Barg, Gerard Cohen, Sylvia Encheva, Grigory Kabatiansky, and Gilles Zemor. A hypergraph approach to the identifying parent property: the case of multiple parents. SIAM J. Disc. Math, 14:423–431, 2001.

[BEN07] Simon Blackburn, Tuvi Etzion, and Siaw-Lynn Ng. Prolific Codes with the Identifiable Parent Property. Cryptology ePrint Archive, Report 2007/276, 2007.

[BEN09] Simon Blackburn, Tuvi Etzion, and Siaw-Lynn Ng. Traceability Codes. Cryptology ePrint Archive, Report 2009/046, 2009.

[BF99] Dan Boneh and Matthew Franklin. An Efficient Public Key Traitor Tracing Scheme. Advances in Cryptology - CRYPTO '99, 1666:783–783, 1999.

[Bla02] Simon Blackburn. An Upper Bound on the Size of a Code with the k-Identifiable Parent Property. Cryptology ePrint Archive, Report 2002/101, 2002.

[Bla03a] Simon Blackburn. An upper bound on the size of a code with the k-identifiable parent property. J. Comb. Theory Ser. A, 102:179–185, 2003.

[Bla03b] Simon Blackburn. Frameproof Codes. SIAM Journal on Discrete Mathematics, 16(3):499–510, 2003.

[BN08] Dan Boneh and Moni Naor. Traitor tracing with constant size ciphertext. Proceedings of the 15th ACM conference on Computer and communications security, pages 501–510, 2008.

[BPS00] Omer Berkman, Michal Parnas, and Jiri Sgall. Efficient dynamic traitor tracing. Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms, pages 586–595, 2000.

[BS98] Dan Boneh and James Shaw. Collusion-Secure Fingerprinting for Digital Data. IEEE Transactions on Information Theory, pages 452–465, 1998.

[BT08] Oded Blayer and Tamir Tassa. Improved versions of Tardos' fingerprinting scheme. Des. Codes Cryptography, 48:79–103, 2008.

[CE00] Gerard Cohen and Sylvia Encheva. Efficient constructions of frameproof codes. Electronics Letters, 36(22):1840–1842, 2000.

[CFNP00] Benny Chor, Amos Fiat, Moni Naor, and Benny Pinkas. Tracing traitors. IEEE Transactions on Information Theory, 46(3):893–910, 2000.

[CKLS96] Ingemar Cox, Joe Kilian, Tom Leighton, and Talal Shamoon. A secure, robust watermark for multimedia. Information Hiding - Lecture Notes in Computer Science, 1174:185–206, 1996.

[EC01] Sylvia Encheva and Gerard Cohen. Some new p-ary two-secure frameproof codes. Applied Mathematics Letters, 14(2):177 – 182, 2001.

[FGC08] Teddy Furon, Arnaud Guyader, and Frederic Cerou. On the Design and Optimization of Tardos Probabilistic Fingerprinting Codes. Information Hiding - Lecture Notes in Computer Science, 5284:341–356, 2008.

[FN94] Amos Fiat and Moni Naor. Broadcast encryption. Proceedings of the 13th annual international cryptology conference on Advances in cryptology, pages 480–491, 1994.

[FPF09] Teddy Furon and Luis Perez-Freire. EM decoding of tardos traitor tracing codes. Proceedings of the 11th ACM workshop on Multimedia and security, pages 99–106, 2009.

[FT01] Amos Fiat and Tamir Tassa. Dynamic Traitor Tracing. Journal of Cryptology, 14:211–223, 2001.

[HM09a] Yen-Wei Huang and Pierre Moulin. Capacity-achieving fingerprint decoding. Information Forensics and Security, 2009. WIFS 2009. First IEEE International Workshop on, pages 51–55, 2009.

[HM09b] Yen-Wei Huang and Pierre Moulin. Saddle-point solution of the fingerprinting capacity game under the marking assumption. Proceedings of the 2009 IEEE international conference on Symposium on Information Theory, 4:2256–2260, 2009.

[HVLLT98] Henk Hollmann, Jack Van Lint, Jean-Paul Linnartz, and Ludo Tolhuizen. On Codes with the Identifiable Parent Property. Journal of Combinatorial Theory, Series A, 82(2):121–133, 1998.

[JKL09] Pascal Junod, Alexandre Karlov, and Arjen Lenstra. Improving the Boneh-Franklin Traitor Tracing Scheme. Proceedings of the 12th International Conference on Practice and Theory in Public Key Cryptography: PKC '09, pages 88–104, 2009.

[JLN04] Hongxia Jin, Jeffery Lotspiech, and Stefan Nusser. Traitor tracing for prerecorded and recordable media. Proceedings of the 4th ACM workshop on Digital rights management, pages 83–90, 2004.

[Ker10] Andrew Ker. The Square Root Law in stegosystems with imprect information. Proc. 12th Information Hiding Workshop - Lecture Notes in Computer Science, 6387:145–160, 2010.

[KSCS07] Stefan Katzenbeisser, Boris Skoric, Mehmet Celik, and Ahmad-Reza Sadeghi. Combining Tardos Fingerprinting Codes and Fingercasting. Information Hiding - Lecture Notes in Computer Science, 4567:294–310, 2007.

[NFH+09] Koji Nuida, Satoshi Fujitsu, Manabu Hagiwara, Takashi Kitagawa, Hajime Watanabe, Kazuto Ogawa, and Hideki Imai. An improvement of discrete Tardos fingerprinting codes. Designs, Codes and Cryptography, 52:339–362, 2009.

[NNL01] Dalit Naor, Moni Naor, and Jeffrey Lotspiech. Revocation and Tracing Schemes for Stateless Receivers. Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology, pages 41–62, 2001.

[Pat07] Maura Paterson. Sequential and dynamic frameproof codes. Des. Codes Cryptography, 42:317–326, 2007.

[PSS03] Chris Peikert, Abhi Shelat, and Adam Smith. Lower bounds for collusion-secure fingerprinting. Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms, pages 472–479, 2003.

[Ron05] Tor Roneid. Collusion-Secure Fingerprinting - A Simulation of the Boneh and Shaw Scheme. Master's Thesis, 2005.

[Sch03] Hans Georg Schaathun. Fighting two pirates. Proceedings of the 15th International Conference on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, pages 71–78, 2003.

[Sch04] Hans Georg Schaathun. Binary collusion-secure codes: Comparison and improvements. Reports in Informatics, University of Bergen, Norway, 2004.

[Sch06] Hans Georg Schaathun. The Boneh-Shaw fingerprinting scheme is better than we thought. IEEE Transactions on Information Forensics and Security, 1:248–255, 2006.

[Sch08] Hans Georg Schaathun. On the Assumption of Equal Contributions in Fingerprinting. IEEE Transactions on Information Forensics and Security, 3:569–572, 2008.

[SDF02] Francesc Sebe and Josep Domingo-Ferrer. Short 3-Secure Fingerprinting Codes for Copyright Protection. Proceedings of the 7th Australian Conference on Information Security and Privacy, pages 316–327, 2002.

[SDF03] Francesc Sebe and Josep Domingo-Ferrer. Collusion-Secure and Cost-Effective Detection of Unlawful Multimedia Redistribution. IEEE Transactions on Systems, Man and Cybernetics, Part C, 33:382–389, 2003.

[SKC08] Boris Skoric, Stefan Katzenbeisser, and Mehmet Celik. Symmetric Tardos fingerprinting codes for arbitrary alphabet sizes. Des. Codes Cryptography, 46(2):137–166, 2008.

[SKSC09] Boris Skoric, Stefan Katzenbeisser, Hans Georg Schaathun, and Mehmet Celik. Tardos fingerprinting codes in the combined digit model. Information Forensics and Security, 2009. WIFS 2009. First IEEE International Workshop on, pages 41–45, 2009.

[SNW03] Reihaneh Safavi-Naini and Yejing Wang. Sequential Traitor Tracing. Proceedings of the 20th Annual International Cryptology Conference on Advances in Cryptology, pages 316–332, 2003.

[SS01] Palash Sarkar and Douglas Stinson. Frameproof and IPP Codes. Progress in Cryptology - INDOCRYPT 2001, 2247:117–126, 2001.

[SS10] Antonino Simone and Boris Skoric. Accusation probabilities in Tardos codes: the Gaussian approximation is better than we thought, 2010.

[SSW01] Jessica Staddon, Douglas Stinson, and Ruizhong Wei. Combinatorial Properties of Frameproof and Traceability Codes. IEEE Transactions on Information Theory, 47:1042–1049, 2001.

[SVCT06] Boris Skoric, Tatiana Vladimirova, Mehmet Celik, and Joop Talstra. Tardos fingerprinting is better than we thought. CoRR, abs/cs/0607131, 2006.

[SW98] Douglas Stinson and Ruizhong Wei. Combinatorial Properties and Constructions of Traceability Schemes and Frameproof Codes. SIAM Journal on Discrete Mathematics, 11(1):41–53, 1998.

[Tar03] Gabor Tardos. Optimal probabilistic fingerprint codes. STOC '03: Proceedings of the thirty-fifth annual ACM symposium on Theory of computing, pages 116–125, 2003.

[Tar09] Gabor Tardos. Tracing traitors - fingerprinting digital documents, 2009.

[Tar10] Gabor Tardos. Capacity of Collusion Secure Fingerprinting - A Tradeoff between Rate and Efficiency. Information Hiding - Lecture Notes in Computer Science, 6387:81–85, 2010.

[Tas05] Tamir Tassa. Low Bandwidth Dynamic Traitor Tracing Schemes. J. Cryptol., 18:167–183, 2005.

[TM05] Tran Van Trung and Sosina Martirosyan. New Constructions for IPP Codes. Des. Codes Cryptography, 35:227–239, 2005.

[XMS07] Yu Xiong, Jun Ma, and Hao Shen. On optimal codes with w-identifiable parent property. Des. Codes Cryptography, 45:65–90, 2007.

[Yac01] Yacov Yacobi. Improved Boneh-Shaw Content Fingerprinting. Proceedings of the 2001 Conference on Topics in Cryptology: The Cryptographer's Track at RSA, pages 378–391, 2001.