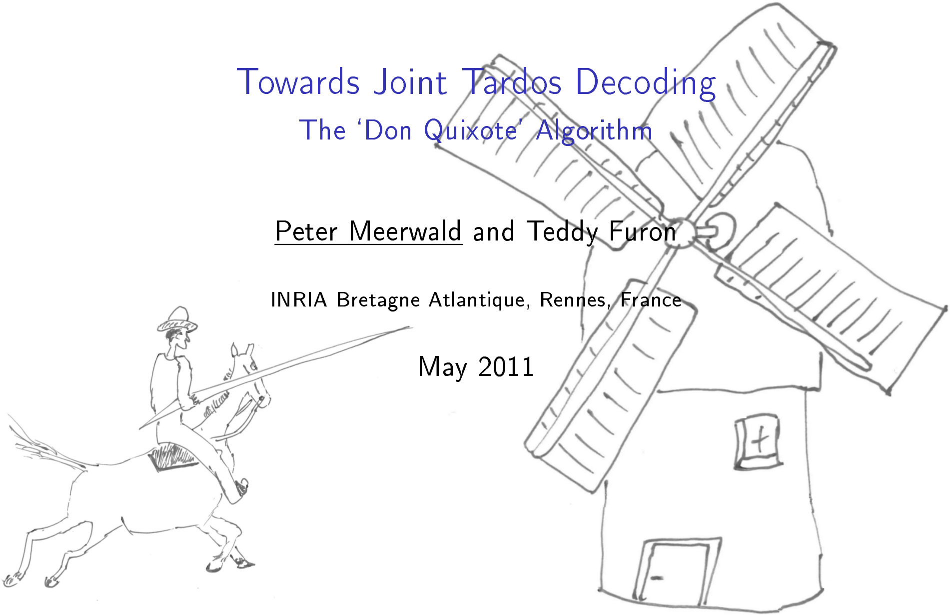
Towards Joint Tardos Decoding

The 'Don Quixote' Algorithm

Peter Meerwald and Teddy Furon

INRIA Bretagne Atlantique, Rennes, France

May 2011



Funded by national project MEDIEVALS ANR-07-AM-005.

Agenda

- Content fingerprinting using Tardos codes
- Iterative, side-informed Tardos decoding
- Inferences about the collusion model
- Making joint decoding affordable - pruning suspects
- Experimental results
  - Detection performance
  - Runtime analysis
- Conclusion

Construction of binary Tardos codes

To support n user, design a binary code matrix $\mathbf{X}$ of size $n \times m$

- Randomly draw m variables $p_i \overset{i.i.d.}{\sim} f(p)$ according to Tardos's arcsine distribution [Tardos, 2003]
- Randomly draw $x_j(i)$ such that $\mathbb{P}(x_j(i) = 1) = p_i$
- Distribute content marked with $\mathbf{x}_j$ to user j

Collusion attack

Colluders $\mathcal{C} = \{j_1, \dots, j_c\}$ forge a pirated copy $\mathbf{y}$ by combining their codewords $\mathbf{x}_{j_1}, \dots, \mathbf{x}_{j_c}$.

$$
\begin{array}{ll}
\mathbf{x}_1 & 0\ 1\ 1\ 0\ 1\ 1\ \dots \\
\mathbf{x}_2 & \boxed{0\ 1\ 0\ 1\ 1\ 0\ \dots} \\
\mathbf{x}_3 & 1\ 0\ 1\ 0\ 0\ 1\ \dots \\
\mathbf{x}_4 & \boxed{0\ 1\ 1\ 1\ 0\ 1\ \dots} \\
\mathbf{x}_5 & 0\ 1\ 0\ 1\ 0\ 1\ \dots \\
\mathbf{y} & \color{red}{0\ 1\ 1\ 1\ 0\ 0\ \dots}
\end{array}
$$

The collusion strategy is denoted $\boldsymbol{\theta}_c = (\theta_c(0), \dots, \theta_c(c))$ with

$$\boldsymbol{\theta}_c(\varphi) = \mathbb{P}(Y = 1 | \sum_{j \in \mathcal{C}} X_j = \varphi).$$

Goal:

- identify one or more colluders given $\mathbf{y}$, $\mathbf{X}$ and $\mathbf{p}$
- maintaining the probability of accusing innocents $< P_{\text{fp}}$

Accusation process

Single decoder: compute score per user

- invariant to collusion attack:

$$s_j = \sum_{i=1}^{m} y(i) \cdot U(x_j(i), p_i) \overset{?}{>} \tau$$ [Skoric et al., 2008]

or

- using an estimate of the collusion:

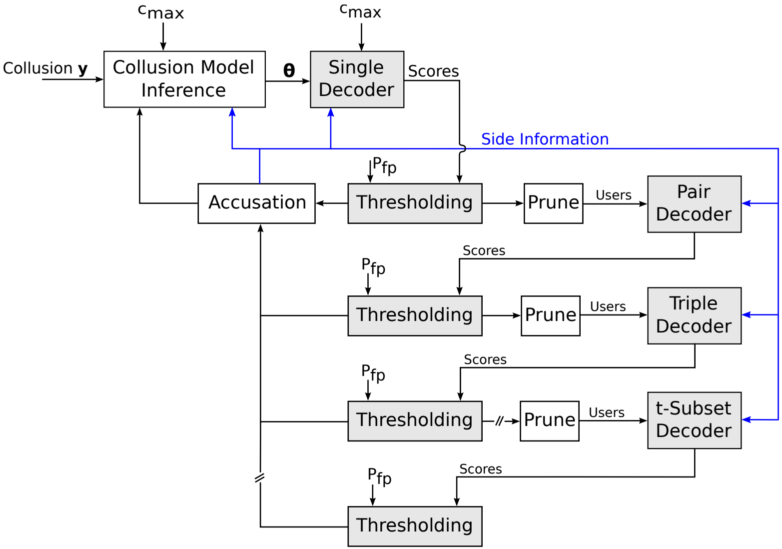
$$s_j = \sum_{i=1}^{m} \log \frac{\mathbb{P}(y(i)|x_j(i), p_i, \hat{\theta}_c)}{\mathbb{P}(y(i)|p_i, \hat{\theta}_c)} \overset{?}{>} \tau$$ [Pérez-Freire & Furon, 2009]

more discriminative, but needs c and accurate $\hat{\theta}_c$

Joint decoder: compute score per subset of t users

- theoretically more discriminative
[Amiri & Tardos, 2009, Moulin, 2008]
- there are $\binom{n}{t}$ user subsets $\rightarrow$ intractable, $O(n^t)$
- limited experimental results for $t = 3$ and $n = 1000$
[Nuida, 2010]

Iterative, side-informed, joint Tardos decoding: Overview



Iterative, side-informed, joint Tardos decoding: Algorithm

Assume $c < c_{\mathsf{max}}$, set side-information $\mathcal{X}_{\mathsf{SI}} = \emptyset$ and repeat until $|\mathcal{X}_{\mathsf{SI}}| \geq c_{\mathsf{max}}$ or $t > t_{\mathsf{max}}$:

1. Infer collusion model $\hat{\boldsymbol{\theta}}$ for c_{max} subject to $\mathcal{X}_{\mathsf{SI}}$
2. Compute score per user (single decoder)
3. Compute accusation threshold τ suject to $\mathcal{X}_{\mathsf{SI}}$ and $\hat{\boldsymbol{\theta}}$ given P_{fp}
4. If scores $> \tau$:
   - 4.1 Accuse user(s) and update side-information $\mathcal{X}_{\mathsf{SI}}$; Go to 1.
5. Set $t = 2$
6. Obtain most likely $p^{(t)}$ user suspects
7. Compute score per suspect subset (joint decoder)
8. Compute accusation threshold τ suject to $\mathcal{X}_{\mathsf{SI}}$ and $\hat{\boldsymbol{\theta}}$ given P_{fp}
9. If top score $> \tau$:
   - 9.1 Accuse most likely suspect in subset and update $\mathcal{X}_{\mathsf{SI}}$; Go to 1.
10. $t = t + 1$ and Go to 6.

Pruning suspects

$O(n^t)$ is intractable $\rightarrow$ limit number of suspects $p^{(t)}$

Assumptions:

- more discriminative scores with each iteration
- likely colluders will move to top of suspect list
- likely innocents get pruned from the suspect list

Subset size (t)	1	2	3	4	6	8
Total subsets $\binom{n}{t}$	10^6	$\sim 10^{11}$	$\sim 10^{17}$	$\sim 10^{22}$	$\sim 10^{33}$	$\sim 10^{43}$
Users suspected $p^{(t)}$	10^6	3000	300	103	41	29
Computed subset scores $\binom{p^{(t)}}{t}$	10^6	$\sim 10^6$	$\sim 10^6$	$\sim 10^6$	$\sim 10^6$	$\sim 10^6$

Score computation of subsets with side-information

The score is the log-likelihood ratio for a user subset $\mathcal{T}$ tuned on the inference $\hat{\boldsymbol{\theta}}_{c_{\max}}$ and side-information $\mathcal{X}_{\mathsf{SI}}$.

$$s_{\mathcal{T}} = \sum_{i=1}^{m} \log \frac{\mathbb{P}(y(i)|\varphi(i), p_i, \hat{\boldsymbol{\theta}}_{c_{\max}}, \rho(i))}{\mathbb{P}(y(i)|p_i, \hat{\boldsymbol{\theta}}_{c_{\max}}, \rho(i))}$$

Accumulated codewords of $\mathcal{X}_{\mathsf{SI}}$ and $\mathcal{T}$:

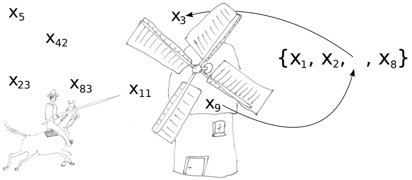
$$\varphi = \sum_{j \in \mathcal{T}} \mathbf{x}_j \text{ and } \rho = \sum_{j \in \mathcal{X}_{\mathsf{SI}}} \mathbf{x}_j$$

The inference $\hat{\boldsymbol{\theta}}_{c_{\max}}$ is not an estimation of the collusion because $c \neq c_{max}$.

$$\hat{\boldsymbol{\theta}}_{c_{\max}} = \underset{\boldsymbol{\theta} \in [0,1]^{c_{\max}+1}}{\arg\max} \log \mathbb{P}(\mathbf{y}|\mathbf{p}, \boldsymbol{\theta}, \mathcal{X}_{\mathsf{SI}}).$$

Implementation Details

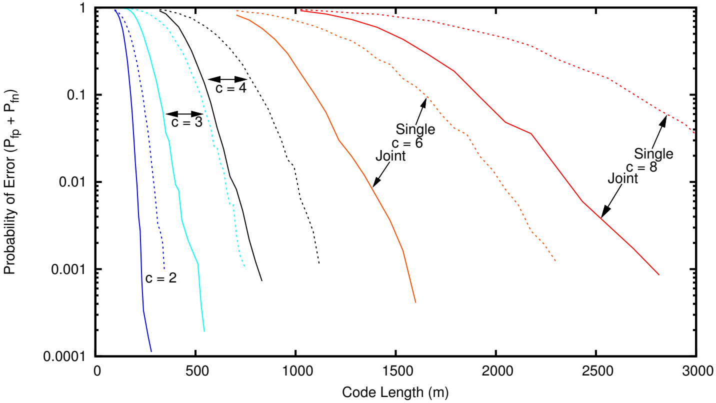
- Implemented decoder in C++, no parallization
  - Fast: can do more than 10^6 scores per second for code length $m = 1024$
  - Runtime results for Intel Core2 CPU (E6700) at 2.6 GHz
- Suspect subsets are enumerated with revolving door algorithm.



- Can use precomputed weights in score computation.

Results: Code length in catch-one scenario (1)

$n = 10^6$, $P_{\text{fp}} = 10^{-3}$, *worst-case* attack



→ Joint decoding reduces required code length.

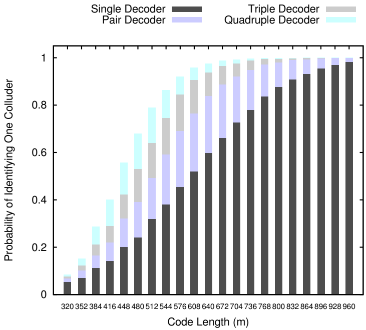
Results: Code length in catch-one scenario (2)

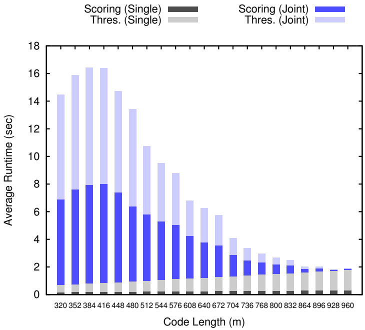
$n = 10^6$, $P_e = 10^{-3}$, *worst-case* attack

Colluders (c)	[Nuida, 2009]	Proposed Decoder Single	Proposed Decoder Joint
2	253	~ 344	~ 232
3	877	~ 752	~ 512
4	1454	~ 1120	~ 784
6	3640	~ 2304	~ 1536
8	6815	~ 3712	~ 2688

Results: Decoder stage making first accusation and runtime

$n = 10^6$, $c = 4$, $P_{\mathrm{fp}} = 10^{-3}$, *worst-case* attack



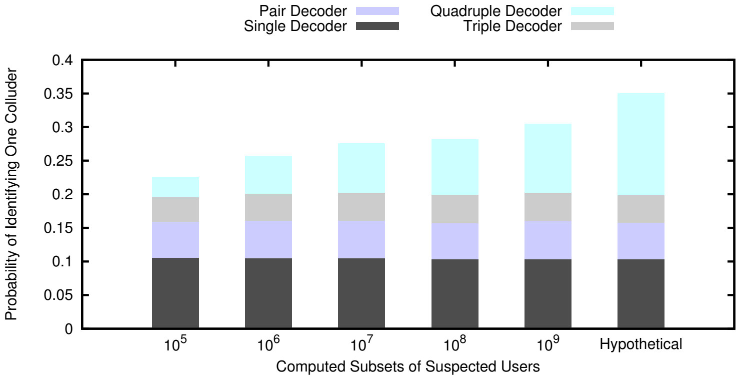


→ Joint decoding improves performance for certain code length with manageable runtime.

Results: Varying number of suspects for joint decoding

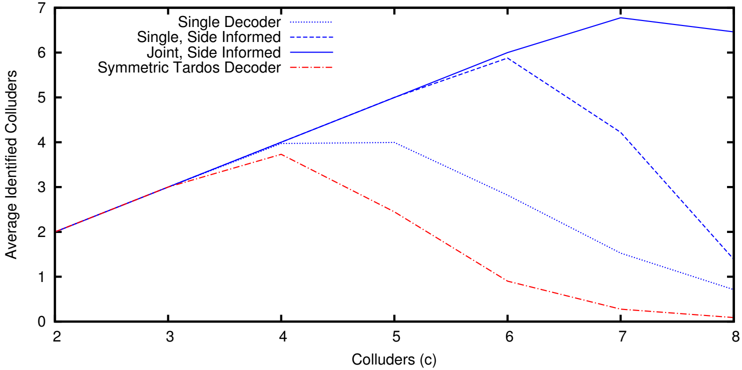
Constraints: $t_{\max} = 4$ and $\binom{p^{(t)}}{t} = 10^5, 10^6, \ldots, 10^9$
Hypothetical: real colluders are never purged

$n = 10^6$, $m = 384$, $c = 4$, $P_{\mathrm{fp}} = 10^{-3}$, *worst-case* attack



Results: Identified colluders in catch-many scenario

$n = 10^6$, $m = 2048$, $P_{\text{fp}} = 10^{-3}$, $c_{\max} = 8$, *worst-case* attack



$\rightarrow$ improvements over symmetric Tardos decoder

Summary

- Focused is on the accusation algorithm
- Thresholding is detailed in the paper: rare-event simulation

In practice what matters is false positive rate of the decoder.

Conclusion

Algorithm for binary Tardos decoding

- main features: practical, joint, scalable
- iterative process: side-information + pruning suspects
- discriminative scores without knowing collusion
- rare event simulation to control false-positive probability

Even small effort in joint decoding increases performance.

AFAIK best decoding performance for binary fingerprinting codes.

Source code available:
`http://www.irisa.fr/texmex/people/furon/src.html`

References

Amiri, E. & Tardos, G. (2009).
High rate fingerprinting codes and the fingerprinting capacity.
In *Proc. ACM-SIAM Sym. on Discrete Algorithms, SODA '09* (pp. 336–345). New York, USA.

Moulin, P. (2008).
Universal fingerprinting: capacity and random-coding exponents.
In *Proc. IEEE Int. Symposium on Inf. Theory* (pp. 220–224). Toronto, ON, Canada.

Nuida, K. (2009).
An improvement of discrete Tardos fingerprinting codes.
Designs, Codes and Cryptography, 52(3), 339–362.

Nuida, K. (2010).
Short collusion-secure fingerprint codes against three pirates.
In *Proc. Information Hiding, IH '10*, volume 6387 of *LNCS* (pp. 86–102). Calgary, Canada.

Pérez-Freire, L. & Furon, T. (2009).
Blind decoder for binary probabilistic traitor tracing codes.
In *Proc. IEEE Int. Workshop on Information Forensics and Security* (pp. 56–60). London, UK.

Skoric, B., Katzenbeisser, S., & Celik, M. (2008).
Symmetric Tardos fingerprinting codes for arbitrary alphabet sizes.
Designs, Codes and Cryptography, 46(2), 137–166.

Tardos, G. (2003).
Optimal probabilistic fingerprint codes.
In *Proc. 35th ACM Sym. on Theory of Computing* (pp. 116–125). San Diego, CA, USA.