技術を社会へ
Integration for Innovation

National Institute of Advanced Industrial Science and Technology
AIST
独立行政法人
産業技術総合研究所

RCIS 情報セキュリティ研究センター
RCIS (Research Center for Information Security)

Short collusion-secure fingerprint codes against three pirates

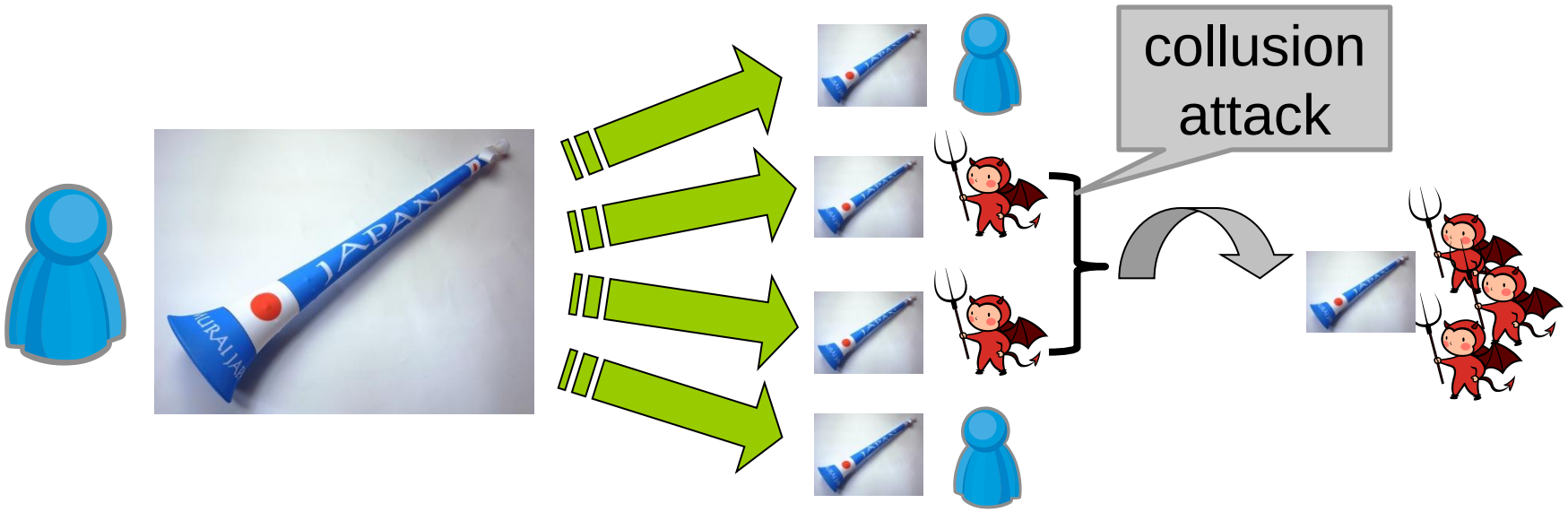
Koji Nuida
RCIS, AIST, Japan

Information Hiding 2010, June 28, 2010

Outline

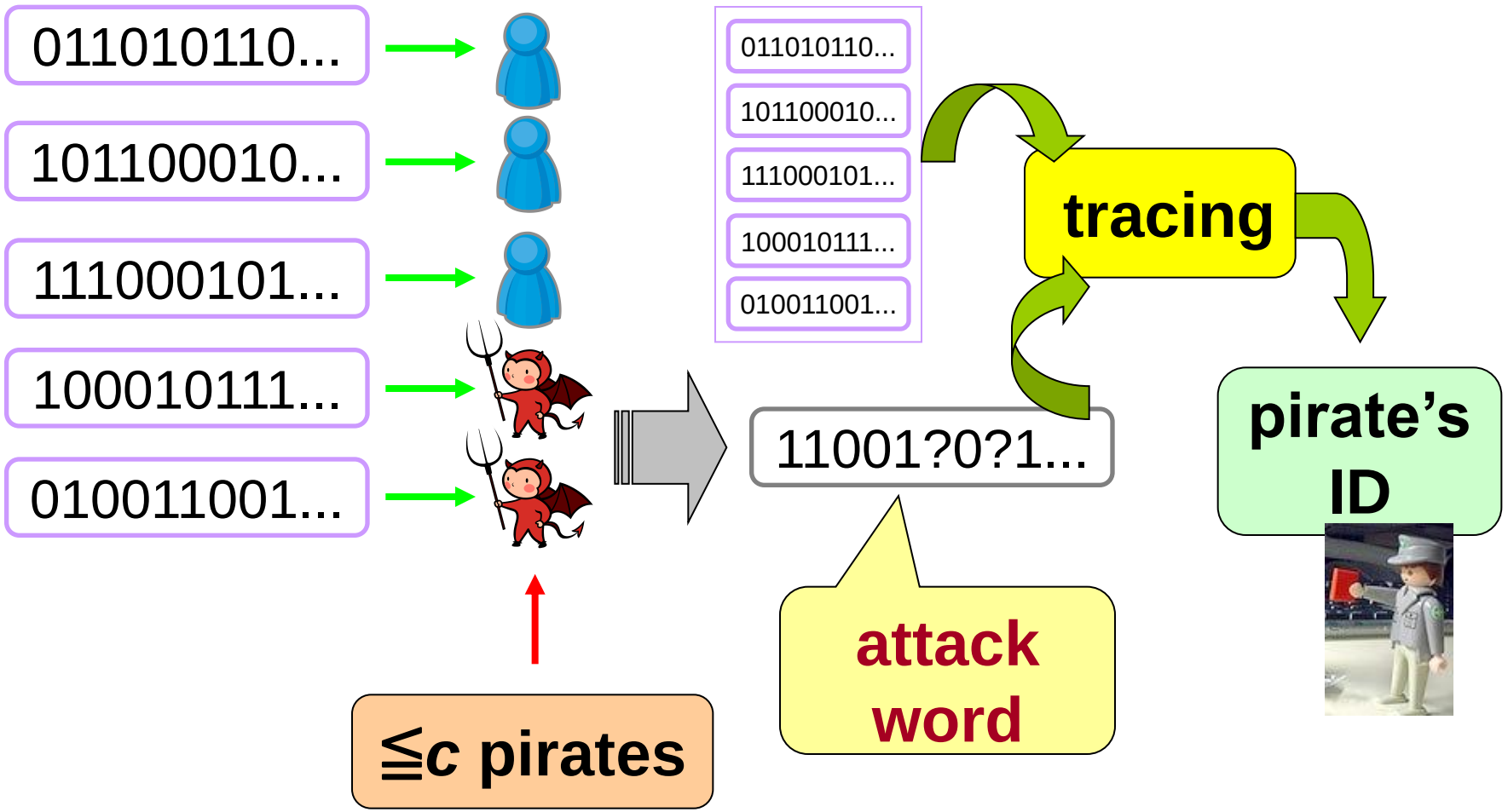
- Problem setting
- Preceding works
- Our contribution: Short 3-secure code
  - E.g., 100 users, 135 bits → 0.9% error
  - Codeword generation (not new)
  - Tracing algorithm (key point)
- Comparison of code lengths
- Observation for speedup of tracing

Problem



- How to prevent illegal redistribution of copied digital content?
  - How to determine the “pirate”?

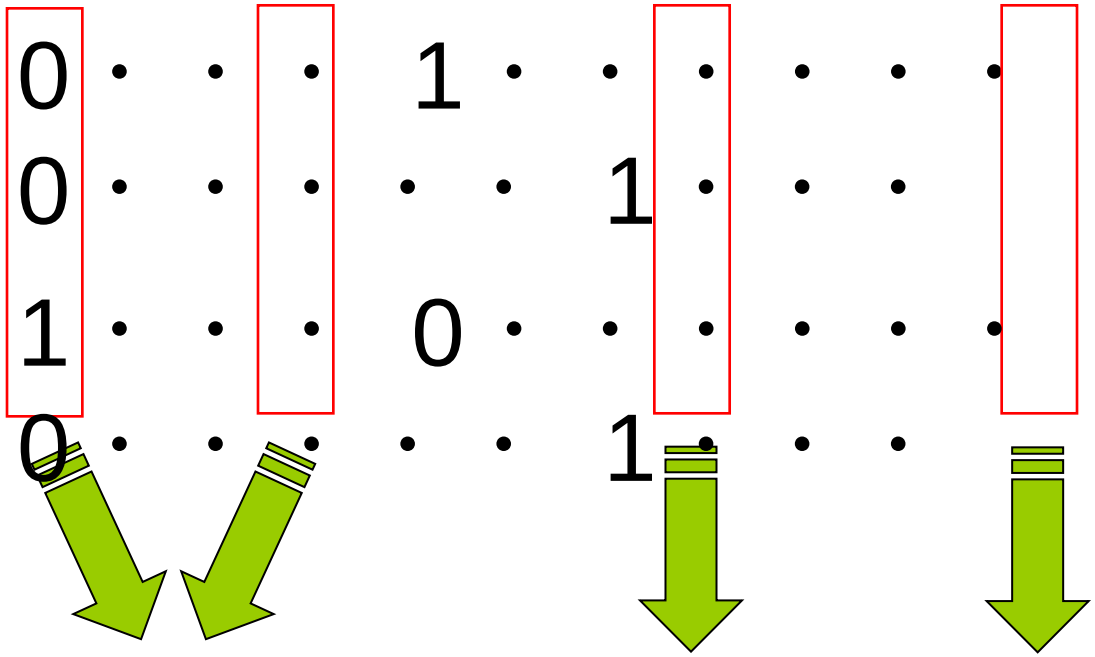
c-Secure Codes



Marking Assumption

[Boneh-Shaw 1995]

“undetectable positions”



0 or 1 or ‘?’ 0 1

Preceding Results

- Tardos' s *c*-secure code [2003]
  - code lengths of optimal order
  - a variant has length asymptotically 1/19 of Tardos [Nuida et al. 2009]
- The shorter, the better
- Further shorter code?
  - e.g. for restricted number *c* of pirates?
    - 2-secure, 3-secure, …

Single or Joint Decoder

- Tracing in Tardos code uses a score for individual user
- Some preceding 2- or 3-secure codes use "parent search" technique
  - Search for a group of users whose codewords can generate the attack word
- More powerful, but less speedy and more difficult to evaluate theoretically

Our Result

- Short 3-secure code with security proof
- Codeword generation is not new
- Tracing algorithm consists of 2 parts
  - 1st part: Score calculation phase
    - Defying “unbalanced” attack strategy
  - 2nd part: Parent search phase
    - Defying “balanced” attack strategy
  - Making the security proof less complex

Codeword Generation

- Each bit of each codeword is chosen uniformly at random
  - Same as Tardos code, but with no bias
- The case of probability $p \neq 0.5$ to choose `1' is also analyzed
- According to the present evaluation, $p = 0.5$ minimizes the "main term" of error probability

Tracing – 1st Phase

- For each codeword *w*, Calculate "(code length) – (Hamming distance of *w* and the attack word)" as score of the user
- Then a user is accused, if the score exceeds a suitably chosen threshold
  - If attack strategy is "unbalanced", then the success probability of this phase becomes higher

Feasible Sets & Parents

- $F(w_1,w_2,w_3)$:= {attack words which can be generated by w_1, w_2 and w_3}
- $T(y)$:= { $\{u_1,u_2,u_3\}$ | y in $F(w_1,w_2,w_3)$ }
- Note: {the 3 pirates} is in T(attack word)

Tracing – 2nd Phase

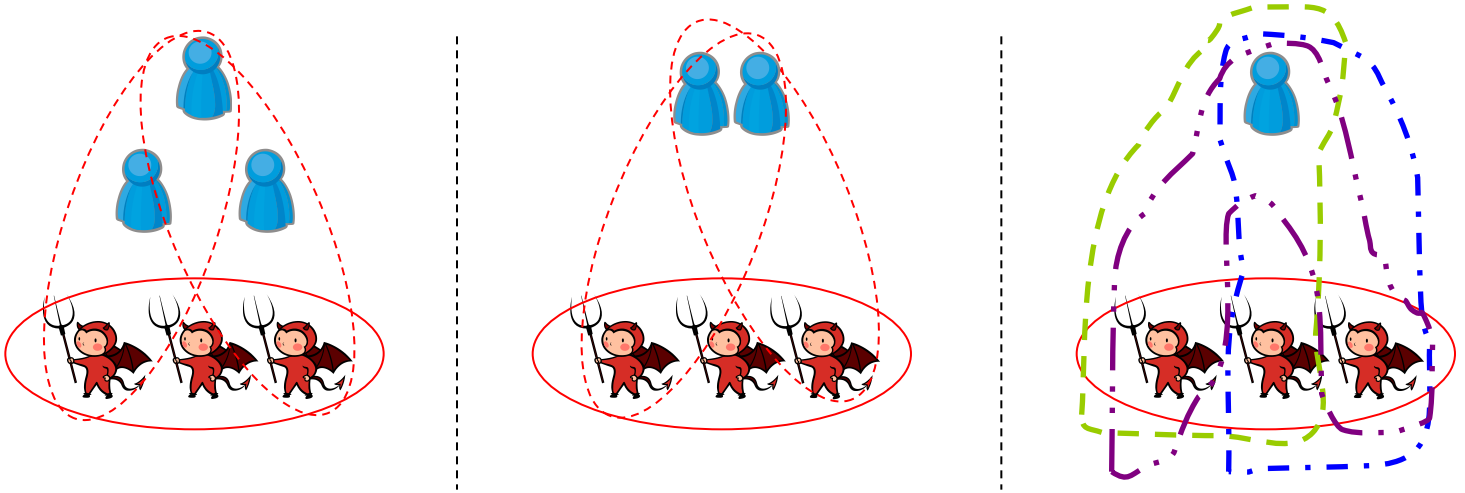
- If $\mathcal{T}' = \{T \in \mathcal{T}(y) \mid T \cap T' \neq \emptyset \ \forall T' \in \mathcal{T}(y)\}$ is empty, then output nobody
- If $\bigcap \mathcal{T}'$ is non-empty, then output its members
- Otherwise, at least one pirate is determined with high probability, by checking the "shape" of $\mathcal{T}'$
  - Thanks to its "asymmetry" (see below)

Some Intuition for 2nd Phase (1)

- The first case is rare, if the code length is sufficiently large
- When the attack strategy is “balanced”, the second step is likely to output some pirate (and no innocent)

Some Intuition for 2nd Phase (2)

- The last step fails only when the following “symmetric” pattern occurs
  - Its probability is negligible, by our analysis



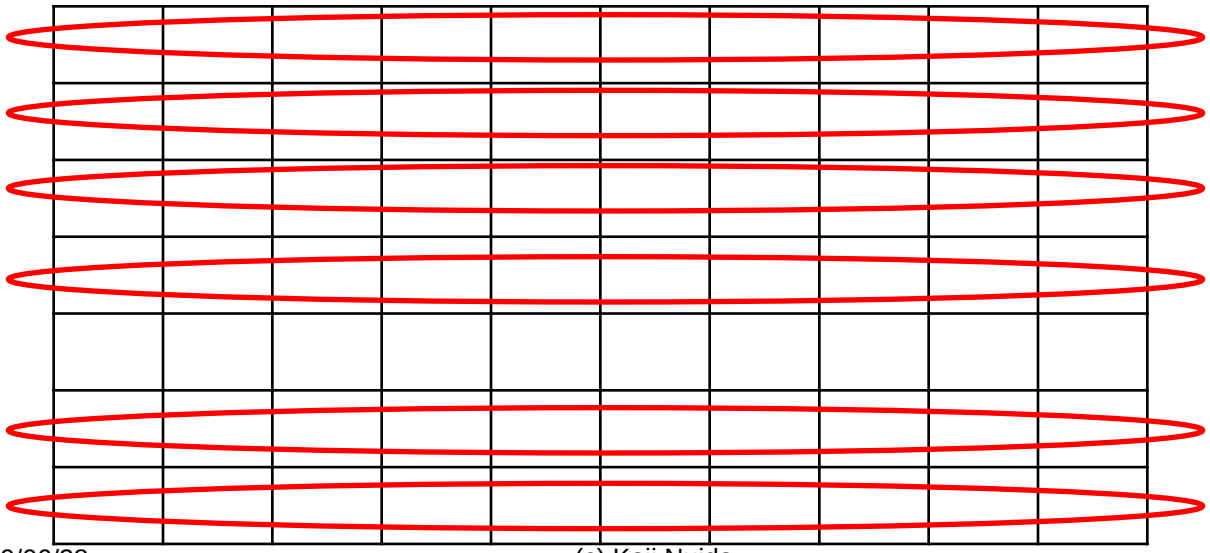
Performance Evaluation

- We gave a formula of error probability
  - The “main term” is about $N^3(7/8)^m/6$
- Example of code lengths

user number N		300	1e+6
error probability		1e-11	1e-3
code length	Nuida ’09	1309	877
	ours	**420**	**349**
ratio		**32.1%**	**39.8%**

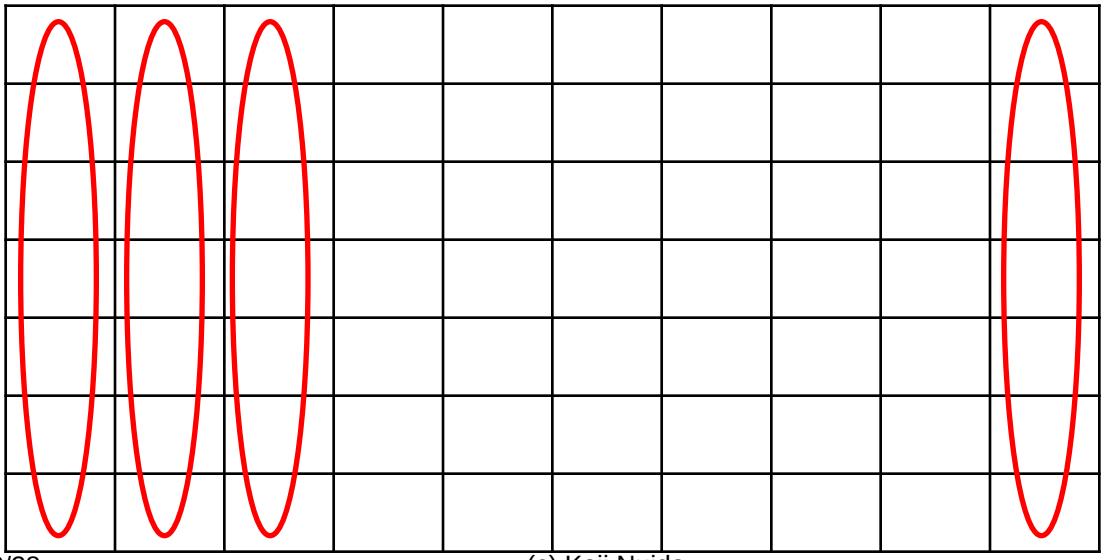
Observation for Speedup (1)

- In a naïve calculation of the set T(y), each row of the codeword matrix is evaluated about $N^2/2$ times



Observation for Speedup (2)

- I tried to evaluate the codeword matrix column-wise, instead of row-wise, to avoid the duplicated evaluation
  - Detailed analysis is future work



Conclusion

- We constructed short 3-secure code, with pirate tracing algorithm combining Tardos's score calculation method with parent search (joint decoding) method
- The code lengths are about 30% to 40% shorter than the existing shortest 3-secure codes
- Speedup of tracing is future work