Randomized Frameproof Codes For Content Protection

N. Prasanth Anthapadmanabhan and Alexander Barg

**Problem Statement**

**Objective**
To design a scheme to protect copyrighted content (e.g., software) against piracy.

**Drawbacks:** Need large redundancy (i.e., low rates), complexity of tracing.

We will consider the following modified problem.

**Definition: Frameproof Code**

What are the rules of the game?
Coalitions try to detect fingerprint positions by comparing their copies.

**Marking assumption** ([Boneh-Shaw 1999])
The coalition can change only those positions of the fingerprint where they find a difference.

**Envelope Set:** Set of all possible forgeries.
\[ E(x_1, \ldots, x_r) = \{ y \mid y = x_i, \forall i \text{ undetectable} \} \]

**Definition**
A randomized code \( C \) is \( t \)-frameproof with \( \varepsilon \)-error if \( \forall U \subseteq [M] \text{ s.t. } |U| \leq t \):

\[
\Pr\left( C(U) \cap C(U') \neq \emptyset \right) \leq \varepsilon.
\]

**Proof Idea:**
For any two pirates, with high prob. the cross-section of their fingerprints contain each of \( (0,0), (0,1), (1,0), (1,1) \) in \( \approx 1/4 \) coordinates.

\[
\begin{align*}
x_1 & = 110010000 \\
x_2 & = 100101010 \\
y & = 110010000
\end{align*}
\]

Given that fingerprints \( (x_1, x_2) \) satisfy above condition:

\[
\Pr\left( (C(U)) \cap (C(U')) \neq \emptyset \right) = \Pr\left( U \subseteq \{x_1, x_2\} \mid (x_1, x_2) \right) \\
= \prod_{i=1}^{2^t} \left( 1 - \frac{1}{2^t} \right) \\
\leq 2^t \cdot 2^{-t} - \varepsilon, \quad 0 \text{ if } t > 2^t.
\]

Do there exist linear \( t \)-frameproof codes for \( t > 2^t \) NO!

**Proposition**
There do NOT exist binary linear \( t \)-frameproof codes with \( \varepsilon \)-error \( 0 \leq \varepsilon < 1, t > 2^t \).

**Binary Frameproof Codes**

**Construction of \( C \):**
Pick random \( M \times n \) binary matrix with \( P(1) = p, P(0) = 1-p \) \( (0 \leq p \leq 1) \).

**Theorem**
\( C \) is \( t \)-frameproof with error probability decaying exponentially in \( n \) for any rate

\[
R < -\varepsilon \cdot \log p - (1-p) \cdot \log (1-p).
\]

**Linear Frameproof Codes**
Linear codes can be validated by verifying parity-checks in \( O(m^2) \)!

**Construction of \( C \):**
Suppose we have \( M = 2^t \) users.

1. Pick random \( m(1-R) \times n \) parity-check matrix with \( P(0) = P(1) = \frac{1}{2} \).
2. Binary vectors satisfying the parity-check matrix form a linear code of size \( \approx 2^m \).
3. Assign to each user a unique codeword selected uniformly at random.

**Theorem**
\( C \) is \( 2 \)-frameproof with error probability decaying exponentially in \( n \) for any rate

\[
R < 0.5
\]

**Notation and Terminology**

- Users: \( [M] = \{ 1, \ldots, M \} \)
- Alphabet: binary
- Fingerprint length: \( n \)
- Rate: \( R = \log \frac{M}{n} \) (quantifies redundancy)
- Collection of fingerprints called a code. A distributor uses randomization, i.e., picks a code at random from a family of codes.
- Randomized code \( C \):
  - Code family \( \{ C_1 \}, C_1 = \{M \} \) - Probability of choosing “key” \( k \) or \( \pi(k) \)
- Validation algorithm:
  - Checks whether fingerprint is present in current code.
  - Preferably polynomial-time complexity.
- Code family known to all users. Distributor keeps selection of \( k \) secret!
- Coalition \( U \subseteq [M] \) (parties of size \( r \) observes \( C(U) = \{x_1, \ldots, x_r\} \). 
- Goals: Distributor: maximize \( R \)
- Coalition: frame an innocent user, i.e., forge \( y, y \in C(x) \cap C(U) \).

**Comparison of Rates**

<table>
<thead>
<tr>
<th>( t )</th>
<th>Randomized</th>
<th>Deterministic</th>
<th>Fingerprinting</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.5</td>
<td>0.2075</td>
<td>0.28</td>
</tr>
<tr>
<td>3</td>
<td>0.25</td>
<td>0.0639</td>
<td>0.0633</td>
</tr>
<tr>
<td>4</td>
<td>0.1392</td>
<td>0.0549</td>
<td>0.0515</td>
</tr>
<tr>
<td>5</td>
<td>0.0666</td>
<td>0.026</td>
<td>0.0006</td>
</tr>
</tbody>
</table>

\( \text{But validation has exponential complexity!} \)

**Polynomial-time validation for larger \( t \)**
We use code concatenation for larger \( t \).

**Corollary**
If \( \frac{\gamma}{\gamma} \geq 1 \) for \( C \) and the error probability \( \varepsilon \), then \( C \) is \( t \)-frameproof with error probability \( 2^n \cdot (2^{-\varepsilon}) \) and has a poly(n) validation algorithm.

**Validation:** Exhaustive search at inner level. Parity-checks at outer level. Choose appropriate scaling, for e.g. \( m \sim \log(M) \) to obtain poly(n) complexity.

**Use explicit codes in the above construction:**
- \( C_{\text{R-S}} \): \( [r-1, K] \) Reed-Solomon code with rate \( (1 - \varepsilon)/r \)
- \( C_{\text{R}} \): randomized binary \( t \)-frameproof with error probability \( e = 2^{-\varepsilon/2} \) for some \( \varepsilon > 0 \), and rate \( R_t \approx \log(\tfrac{1}{1-p} \log(1-p) / \log(1-p)) \).

Taking \( \varepsilon \) arbitrarily small and \( n \) sufficiently large to satisfy \( e < \varepsilon \), we obtain:

**Theorem**
The concatenated code is \( t \)-frameproof with error prob. \( \exp(-\tfrac{1}{n}(\varepsilon)) \), validation complexity \( O(n^2) \) and rate \( R_t/t \).