Non-Malleable Codes from Two-Source Extractors

Stefan Dziembowski
Tomasz Kazana
Maciej Obremski

University of Warsaw
New trend in cryptography

construct schemes secure against the “physical attacks"
Physical attacks

1. Information leakage
   (side-channel attacks)

2. Malicious modifications
   (tampering attacks)

MACHINE
(PC, smartcard, etc.)
Basic question

How to construct encoding schemes secure against the physical attacks?

Encoding scheme \( (\text{Enc}: \mathcal{M} \rightarrow C, \text{Dec}: C \rightarrow \mathcal{M}) \) such that:
- \( \text{Enc} \) can be randomized
- \( \text{Dec}(\text{Enc}(M)) = M \)

If this encoding has some “homomorphic” properties then one could do computation with encoded values.
How to define encoding schemes secure against leakage?

[DDV10]:

\[ H \] – some family of functions

- Obviously:
  - \( H \) needs to be restricted in some way
  - e.g. the identity function cannot be in \( H \)

\[ C = \text{Enc}(M) \]

chooses \( h \in H \)

learns \( h(C) \)

M?
Lots of variants

- probing

- **bounded** leakage (sometimes in the “**split-state model**”)

- **noisy** leakage

- **computationally-restricted** leakage
Bounded leakage in the **split-state model**

\[ Enc(M) = (L, R) \]

- \( f \) such that \(|f(L)| \ll |L|\)
- \( g \) such that \(|g(R)| \ll |R|\)

learns \( f(L), g(R) \)
Split-state model - advantages

• Easily implementable

• **Generalizes** some other models (probing, “only computation leaks”…)

• **Useful** when the adversary shouldn’t get access to full encoding

• Related to the theory of **two-source extractors**.
How to construct encoding resilient to such leakage?

Use two-source randomness extractors.

Recall:
\[ \text{ext}: \mathcal{L} \times \mathcal{R} \rightarrow \mathcal{M} \text{ is a two-source randomness extractor if:} \]
for \( L \) and \( R \) having large min-entropy

\[ \text{ext}(L',R') \text{ is close to uniform} \]

(in strong extractors this holds even if one learns \( L \) or \( R \)).
Min-entropy

\[ H_\infty(X) = - \log_2 \max_x P(X = x) \]

**Example:**

\[ H_\infty(X) = a \text{ if } X \text{ is distributed uniformly over set of size } 2^a \]
How to encode a message?

\[
\text{ext: } \mathcal{L} \times \mathcal{R} \rightarrow \mathcal{M} \text{ --- a two-source randomness}
\]

Construct:

\[
(\text{Enc: } \mathcal{M} \rightarrow \mathcal{L} \times \mathcal{R}, \text{ Dec: } \mathcal{L} \times \mathcal{R} \rightarrow \mathcal{M})
\]

by first defining the **decoding** as:

\[
\text{Dec}(L,R) := \text{Ext}(L,R)
\]

And then the **encoding** as:

\[
\text{Enc}(M) \leftarrow \{(L,R) : \text{Dec}(L,R) = M\}
\]
Why does it work?

Since

$$|f(L)| \ll |L|$$

hence:

most likely

$$\mathcal{L}' = \{L : f(L) = \text{out}\}$$

is large

where $\text{out} := f(L)$

(same for $g$ and $R$)

Hence: $f(L)$ and $g(R)$ don’t give much information on $M$
A popular two-source extractor: the inner product

\[ F \text{ – finite field, } n \geq 2 \]

Let

\[ \mathcal{L} = \mathcal{R} = F^n \]

Then

\[ \operatorname{Ext}((L_1, \ldots, L_n), (R_1, \ldots, R_n)) := \langle (L_1, \ldots, L_n), (R_1, \ldots, R_n) \rangle \]

is a strong two-source extractor with good parameters.
Plugging this extractor into our encoding:

\[ F – \text{finite field, } n \geq 2 \]

\[ \text{Enc}(M) := \text{random } (L,R) \text{ such that } \langle L,R \rangle = M \]

\[ \text{Dec}(L,R) := \langle L,R \rangle \]
What about encoding resilient to tampering?

Less clear how to define it...

Consider the following tampering experiment:

\[
\begin{align*}
  M &\xrightarrow{\text{Enc}} C = \text{Enc}(M) &\xrightarrow{\text{Dec}} M' = \text{Dec}(M) \\
  C' &= h(C) \\
\end{align*}
\]

chooses \( h \in \mathcal{H} \)

induces:

\[
\begin{align*}
  M &\xrightarrow{h'} M' = \text{Dec}(M) \\
\end{align*}
\]
What functions can the adversary induce?

Even for very restricted families $\mathcal{H}$ he can

- make $h'(M) = M$
- make $h'(M) = \text{constant } X$ “independent from } M”

choosing $h(C) = C$

or

choosing $h(C) = \text{Enc}(X)$
Non-Malleable Codes [DPW10]

The “identity” and the “constant” attacks should be the only thing that the adversary can do.

A bit more formally:

\((\text{Enc},\text{Dec})\) is non-malleable with respect to family \(\mathcal{H}\) if \(h'\) can be represented as a probabilistic combination of:

- the \textit{identity} function
- \textit{constant} functions
How to formalize that $h'$ is a probabilistic combination of constant functions?

$$(	ext{Enc: } M \rightarrow C, \text{ Dec: } C \rightarrow M) \text{ is } \epsilon\text{-non-malleable w.r.t. } \mathcal{H} \text{ if}$$

$$\forall h \in \mathcal{H} \quad \exists D \text{ – distribution on } M$$

such that

$$\forall M \in \mathcal{M} \quad \text{Dec}(h(\text{Enc}(M))) \approx_{\epsilon} D$$

**Problem:** what to do with the “identity” functions?
Solution

\( D \) – distribution on \( \mathcal{M} \cup \{ \text{same} \} \)

For \( M \in \mathcal{M} \) define:

\[
\text{Tamper}_M(D) :=
\]

1. sample \( x \leftarrow D \)
2. if \( x = \text{same} \) then output \( M \)
3. otherwise output \( x \)

\((\text{Enc}, \text{Dec})\) is \( \epsilon \)-non-malleable w.r.t. \( \mathcal{H} \) if

\[
\forall h \in \mathcal{H}, \quad \exists D, \quad \forall M \in \mathcal{M} \quad \text{such that} \quad \forall \text{Dec}( h( \text{Enc}(M) ) ) \approx^\epsilon \text{Tamper}_M(D)
\]
An example of an application: related key attacks

If the key after tampering is either identical or independent then tampering is useless for the adversary.
D., Pietrzak and Wichs (ICS 2010)

• **Existence** of non-malleable codes for small enough families $\mathcal{H}$ via **probabilistic argument**

• Explicit construction of non-malleable codes with respect to **independent bit tampering**
Independent Bit Tampering

4 types of functions acting on bits:

- **Keep**: The bit remains unchanged.
- **Flip**: The bit is inverted (0 becomes 1, and 1 becomes 0).
- **Set to 0**: The bit is set to 0.
- **Set to 1**: The bit is set to 1.

<table>
<thead>
<tr>
<th>C_1</th>
<th>C_2</th>
<th>C_3</th>
<th>C_4</th>
<th>C_5</th>
<th>C_6</th>
<th>C_7</th>
<th>C_8</th>
<th>C_9</th>
<th>C_10</th>
</tr>
</thead>
<tbody>
<tr>
<td>keep</td>
<td>set to 0</td>
<td>flip</td>
<td>keep</td>
<td>set to 0</td>
<td>flip</td>
<td>set to 1</td>
<td>flip</td>
<td>set to 1</td>
<td>keep</td>
</tr>
<tr>
<td>C_1</td>
<td>0</td>
<td>neg C_3</td>
<td>C_4</td>
<td>0</td>
<td>1</td>
<td>neg C_7</td>
<td>1</td>
<td>neg C_9</td>
<td>M_{10}</td>
</tr>
</tbody>
</table>
Non-malleable code secure against independent bit tampering

[DPW10]:
A construction of an efficient non-malleable code secure against independent bit tampering.

It achieves the rate of \( \approx 0.1887 \).

Uses the algebraic manipulation detection codes [CDFPW08].
An interesting open problem from [DPW10]

Construct a non-malleable code secure in the split-state model.

More precisely suppose that $\text{Enc}(M) = (L, R)$

$(f, g)$ – arbitrary tampering functions $f$ and $g$ are applied separately to $L$ and $R$:

Motivation:
- easily implementable in practice
- well-studied model in the leakage-resilient crypto
- generalizes some other models (e.g. the independent bit tampering)
Progress towards solving this problem

1. [DPW10]: existence result

2. [Liu and Lysyanskaya, Crypto 2012]:
   
   – explicit construction, computational-security, assuming common reference string
   
   – bonus feature: resilient to leakage.
Our result

• explicit, non-malleable code for 1-bit messages in split-state model

• also resilient to leakage
Fact

Scheme \((\text{Enc} : \{0,1\} \rightarrow C, \text{Dec}: C \rightarrow \{0,1\})\) is \(\varepsilon\)-non-malleable with respect to family \(\mathcal{H}\) if for every \(h \in \mathcal{H}\):

\[
P(M \neq h'(M)) \leq 1/2 + \varepsilon
\]

Recall:
\[
h'(M) = \text{Dec}(h(\text{Enc}(M)))
\]

In other words:
\[
P(h'(1)=0) + P(h'(0)=1) \leq 1 + \varepsilon
\]

where \(M\) is uniformly distributed over \(\{0,1\}\).
hard to negate $\Rightarrow$ non-malleable

$P(h'(1)=0) + P(h'(0)=1) \leq 1$

look at the distributions of:

$h(0)$:
- probability of 0
- probability of 1

$h(1)$:
- probability of 0
- probability of 1

$D$:
- probability of 0
- same
- probability of 1
non-malleable $\Rightarrow$ hard to negate

Distributions of $h(0)$:
- Probability of 0
- Probability of 1

$P(0 = h'(1))$

$D$:
- Probability of 0
- Same
- Probability of 1

$P(h'(1) = 0)$

$P(h'(1) = 0) + P(h'(0) = 1) \leq 1$

Hence: $P(h'(1) = 0) + P(h'(0) = 1) \leq 1$
Look again at our problem:

\[ M \leftarrow \{0,1\} \]

\[ \text{Enc}(M) \]

\[ M' := \text{Dec}(L',R') \]

Goal: construct encoding where \( P(M' \neq M) \leq 1/2 \)
Observation 1

Enc needs to be a 2-out-of-2 secret sharing.

Why?

Suppose L reveals some information about M. This means that there are two sets

\[ L_0 \text{ and } L_1 \]

such that for \( m \in \{0,1\} \) if \( L \in L_m \) then M is more likely to be equal to \( m \).

Then Alice can “negate” M by swapping elements in \( L_0 \) and \( L_1 \).
Not every secret sharing works!
For example: take $\mathcal{L} = \mathcal{R} = \mathbb{Z}_2$ and define:
\[
\text{Enc}(M) \leftarrow \{(L,R) = L + R = M \mod 2\}
\]

\[
L' := L + 1 \mod 2
\]

\[
M' := L' + R = M + 1 \mod 2
\]
Hence:

we need to use secret sharing

**with some additional properties**

**Idea**: use extractors exactly as in the case of leakage-resilient encoding.

**(hope**: this should give us also some leakage-resilience)
Let’s see if it works:

\( F \) – finite field, \( n \geq 2 \)

\[
\text{Enc}(M) := \text{random } (L,R) \text{ such that } \langle L,R \rangle = M
\]

\[
\text{Dec}(L,R) := \langle L,R \rangle
\]

**Problem**: linearity of the inner product (let \( c \in F \))

\[
\langle c \cdot L,R \rangle = c \cdot \langle L,R \rangle
\]

**So**: if we choose

\( f(L) = c \cdot L \) and \( g(R) = r \)

then \( M' = c \cdot M \)
Observation 3

If $F = Z_2$ then the $c$ can only be 0 or 1

- if $c = 0$ then it is a "constant attack":
  \[ M' = 0 \text{ for every } M \]

- if $c = 1$ then it is an "identity attack":
  \[ M' = M \text{ for every } M \]
Hope: maybe it works over $\mathbb{Z}_2$

Unfortunately here another attack is possible:

Note: the inner product changes iff $L_1R_1 = 0$.
This happens with probability $\frac{3}{4}$. 
Observation 4

The attack from the previous slide does not work if $|F|$ is exponential.

This is because

$$P(L_1 R_1 \neq 0) \approx (1 - 1/|F|)^2$$
So, this is the situation

<table>
<thead>
<tr>
<th></th>
<th>large $F$</th>
<th>$F = {0,1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>the “linear attack”</td>
<td>works</td>
<td>doesn’t work</td>
</tr>
<tr>
<td>the “$L[1] := 1$ and $R[1] = 1$” attack</td>
<td>doesn’t work</td>
<td>works</td>
</tr>
</tbody>
</table>

**Question**: is it possible to combine these two solutions so that none of these attacks works?
Define

\[(\text{Enc} : \mathbb{Z}_2 \rightarrow F^n \times F^n , \text{Dec}: F^n \times F^n \rightarrow \mathbb{Z}_2)\]

as

\[\text{Enc}(M) := \begin{cases} 
\text{random (L,R) such that } \langle L, R \rangle = 0 \text{ if } M = 0 \\
\text{random (L,R) such that } \langle L, R \rangle \neq 0 \text{ if } M = 1
\end{cases} \]

\[\text{Dec}(L,R) \text{ just computes } \langle L, R \rangle \text{ and checks if it is 0.} \]

\textbf{In fact}: the inner product can be replaced by \textbf{any sufficiently strong two-source extractor} with large output
Proof: general idea

First observe that:

If $f$ or $g$ are both constant then obviously $M'$ is not correlated with $M$.

Actually: it’s enough if one of $f$ and $g$ is constant.

Or even: if it “glues” sufficiently many inputs.
Why is this?

“gluing” ≈ “forgetting”

from “strongness” of the extractor:

even if one learns $\mathbb{R}$

the fact that

$L \in \mathcal{L}'$

doesn’t reveal anything about $\langle L, \mathbb{R} \rangle$
So, suppose $f$ and $g$ don’t glue too many elements together

But in this case: $f(L)$ and $g(R)$ have a lot of min-entropy.

This fact can be used to show that if $\langle L, R \rangle \neq 0$ then it is very unlikely that $\langle f(L), (R) \rangle = 0$.

**Informally**: “it is hard to go $1 \rightarrow 0$”
Putting it together

The adversary can apply a “mix” of strategies:

• “close to constant” – in this case the probability of negating is around $1/2$.
  or
• “far from constant” – in this case the probability of negating 1 is close to 0, so the total probability of negating $M$ is also at most $1/2$.

this finishes the proof sketch.
Bonus feature – leakage-resilience

The adversary before choosing the tampering functions can adaptively leak from the codeword:

\[ \text{total leakage} < \frac{|L|}{2} \]
A recent exciting paper

Aggarwal, Dodis, and Lovett *Non-malleable Codes from Additive Combinatorics*.

General outline of their method:
1. show that mauling the inner product encoding can induce only *affine functions* $h'$ (or their random combinations)
2. on top of it use encoding that is *resilient to affine mauling*. 
Thank you!