Efficient Non-Malleable Codes and Key-derivations against Poly-size Tampering Circuits

PRATYAY MUKHERJEE
(Aarhus University)
Joint work with
Sebastian Faust, Daniele Venturi and Daniel Wichs

EUROCRYPT 2014, COPENHAGEN

May 12, 2014
Two Parts

Part-1
Efficient Non-malleable Codes
against Poly-size circuits

Part-2
Efficient Non-malleable Key-derivation
against Poly-size circuits
Part-1

Efficient Non-malleable Codes against Poly-size circuits

Non-malleable Codes (Informally)
A modified codeword contains either original or unrelated message.

E.g. Can not flip one bit of encoded message by modifying the codeword.
The “Tampering Experiment”

Consider the following experiment for some encoding scheme \((\text{ENC},\text{DEC})\)

\[ C \xrightarrow{\text{ENC}} S \xrightarrow{\text{Tamper}} C^* = f(C) \xrightarrow{\text{DEC}} S^* \]

**Note**
- ENC can be randomized.
- There is no secret Key.

**Goal:**
Design encoding scheme \((\text{ENC},\text{DEC})\) which is **Non-malleable** for an “interesting” class \(I\)
**Tamper}^f(s)**

\[
\text{ENC} \quad \rightarrow \quad C \quad \rightarrow \quad \text{Tamper} \quad \rightarrow \quad C^* = f(C) \quad \rightarrow \quad \text{DEC} \quad \rightarrow \quad s^*
\]

\[f \in \mathcal{I} \quad \#\]

If \(C^* = C\) return **same** Else return \(s^*\)

**Definition [DPW 10]:**

A code \((\text{ENC}, \text{DEC})\) is **non-malleable** w.r.t. \(\mathcal{I} \quad \#f\); \(f \in \mathcal{I}\) and \(s_0, s_1\) we have:

\[
\text{Tamper}^f(s_0) \approx \text{Tamper}^f(s_1)
\]
Application : Tamper-Resilient Cryptography

• Non-malleable codes are used to protect against key-tampering attacks.

• How ?
  – Encode the key using NMC.
  – The tampering adversary can not modify the encoded key to some related key.
Limitation and Possibility

**Limitation:** For any (ENC, DEC) there exists $f_{bad}$ which decodes $C$, flips 1-bit and re-encodes to $C^*$.  

**Corollary-1:** It is impossible to construct encoding scheme which is non-malleable w.r.t. all functions $I_{all}$.  

**Corollary-2:** It is impossible to construct efficient encoding scheme which is non-malleable w.r.t. all efficient functions $I_{eff}$.  

**Question:** How to restrict $I_{all}$?  

**Way-1:** Restrict granularity  
- Codeword consists of components which are independently tamperable.  
- **Example:** Split-state tampering  
  $[DPW10, LL12, DKO13, ADL13, CG13, FMNV13, ADK14]$:

**Way-2:** Restrict complexity  
- The whole codeword is tamperable but only with functions that are not “too complicated”.

Our Focus!
Our Result

Corollary-2: It is impossible to construct efficient encoding scheme which is non-malleable w.r.t. all efficient functions $I_{\text{eff}}$.

Main Result: “The next best thing”
For any fixed polynomial $P$, there exists an efficient non-malleable code for any family of functions $|I| \leq 2^P$.

Corollary-3
For any fixed polynomial $P$, there exists an efficient non-malleable code for all circuits of size $\leq P$. 
Our Result

A similar result [CG 14]
But the encoding/decoding becomes “inefficient” in order to get negligible error

Main Result: “The next best thing”
For any fixed polynomial $P$, there exists an efficient non-malleable code for any family of functions $|\mathcal{I} \#| \leq 2^P$.

Corollary-3
For any fixed polynomial $P$, there exists an efficient non-malleable code for all circuits of size $\leq P$.

Caveat: Our results hold in CRS model.
NMC in CRS model

- Fix some polynomial $P$

- We construct a family of efficient codes parameterized by $\text{CRS}: (\text{ENC}_{\text{CRS}}, \text{DEC}_{\text{CRS}})$

- We show that, w.h.p. over the random choice of $\text{CRS}$: $(\text{ENC}_{\text{CRS}}, \text{DEC}_{\text{CRS}})$ is an NMC w.r.t. all tampering circuits of size $\leq P$

Although $P$ is chosen apriori, the tampering circuit can be chosen from the family of all circuits of size $\leq P$ adaptively.
The Construction Overview

Input: \( s \)

**Intuitions (outer encoding)**

Ingredient: a \( t \)-wise independent hash function \( h \)

\[
D \overset{\text{\(=\)}}{=} \ D_2 \ || \ h(\ D_2 \ )
\]

\( D \) is Valid \( \iff \) \( D \) is of the form \( S \ || \ h(\ S \ ) \)

\begin{itemize}
  \item We choose \textbf{CRS} such that \( |\text{Circuit computing } h| > P \)
  \Rightarrow No circuit of size \( \leq P \) can compute \( h \) on “too many” points. (Proof: Probabilistic Method)
  \item For every tampering function \( f \) there is a “small set” \( S_f \) such that if a tampered codeword is valid, then it is in \( S_f \) w.h.p.
\end{itemize}
The Construction Overview

Input: \( s \)

Inner Encoding

\( D_2 \)

Outer Encoding

\( C \)

Intuitions (outer encoding)

- For every tampering function \( f \) there is a “small set” \( S_f \) such that if a tampered codeword is valid, then it is in \( S_f \) w.h.p.

We call this property **Bounded Malleability** which ensures that the tampered codeword does not contain “too much information” about the input codeword.
The Construction Overview

Input: \( s \)

**Intuitions (Inner encoding)**

- **Output of** \( \text{Tamper}^f(s) \) **can be thought of as some leakage on** \( C_1 \). w.h.p. the leakage range is “small”: \( \{\text{same, } \perp, S_f\} \)

**Tamper^f(s)**

\[
\begin{align*}
S & \xrightarrow{\text{ENC}} C \xrightarrow{\text{Tamper}} C^* = f(C) \xrightarrow{\text{DEC}} S^*
\end{align*}
\]

- If \( C^* = C \) return **same** Else return \( S^* \)

**Example**

- \( f \) can guess some bit(s) of \( D_2 \) and if the guess is correct, leave \( D \) **same** otherwise overwrites to some invalid code.
**Leakage-Resilient Code**

**Def [DDV 10]:** A code $(\text{LRENC}, \text{LRDEC})$ is **leakage-resilient** w.r.t. $J$ if

$$g \in J \quad \text{and} \quad s: \quad g(\text{LRENC}(s)) \approx g(U)$$

**Construction [DDV 10]:** Let $h'$ be a $t$-wise hash function. Then to encode $s$, choose a random $r$ and output $c = r || h'(r) \oplus s$

**Analysis by [DDV 10]:** The construction is an LRC as long as:

$$r > \ell \quad \text{even if} \quad r << s$$

We show: The construction is an LRC as long as:

$$r > \ell \quad \text{even if} \quad r << s$$

We use the same construction but improved analysis to achieve optimal rate $\approx 1$. 

Our Inner Encoding
Putting it together

Input: $s$

Inner Encoding

$D_2$

Outer Encoding

Leakage Resilient Code

Bounded Malleable Code

Non-Malleable Code
Part-2
Efficient Non-malleable **Key-derivation (NMKD)** against Poly-size circuits
NMKD: A new primitive

Source: \( X \)

Tampered Source: \( f(X) \)

NMKD guarantees that if \( f(X) \neq X \) then \( (Y, Y') \approx (U, Y') \)

A dual of Non-Malleable Extractor
**NMKD: Definition**

**Real** $\phi, f$

Sample $x \leftarrow U$

If $f(x) = x$

return $(\phi(x), \text{same})$

Else return $(\phi(x), \phi(f(x)))$

**Ideal** $\phi, f$

Sample $x \leftarrow U$ ; $y \leftarrow U'$

If $f(x) = x$

return $(y, \text{same})$

Else return $(y, \phi(f(x)))$

**Definition:** A function $\phi$ is **NMKD** w.r.t. $\mathcal{I}$ if

$f \in \mathcal{I}$ if above holds

**Theorem (NMKD):**

For any $\mathcal{I}$ of size $\leq 2^p$, a randomly chosen $t$-wise independent hash function is an NMKD w.h.p. as long as $t > P$
Conclusion

• The first construction of non-granular efficient Non-malleable code.
  – Our construction is information theoretic and achieves optimal rate.

• A new primitive Non-Malleable Key-derivation.
  – Application to construct Tamper-resilient Stream Cipher.

• Open:
  – New Application of NMKD.
Thank You!