Non-malleable codes in the split-state model

Divesh Aggarwal, **Yevgeniy Dodis**, Tomasz Kazana,
Shachar Lovett, Maciej Obremski

New York University

Tampering Experiment

$$c \xrightarrow{f} c^*$$

- Consider a tamperable communication channel.

Tampering Experiment

$$m \xrightarrow{\text{Enc}} c \xrightarrow{f} c^*$$

- Consider a tamperable communication channel.
- To protect, send $c = \text{Enc}(m)$ along the channel.

Tampering Experiment

$$m \xrightarrow{\text{Enc}} c \xrightarrow{f} c^* \xrightarrow{\text{Dec}} m^* \qquad \textit{(Real)}$$

- Consider a tamperable communication channel.
- To protect, send $c = \text{Enc}(m)$ along the channel.
- The tampered codeword decodes to some m^*.

Tampering Experiment

$$m \xrightarrow{\text{Enc}} c \xrightarrow{f} c^* \xrightarrow{\text{Dec}} m^* \qquad \textit{(Real)}$$

$$\approx \quad m \xrightarrow{\quad\quad g \quad\quad} g(m) \qquad \textit{(Ideal)}$$

- Consider a tamperable communication channel.
- To protect, send $c = \text{Enc}(m)$ along the channel.
- The tampered codeword decodes to some m^*.
- Hope: m^* "looks like" $g(m)$ for some "good" g that we can "tolerate".

Tampering Experiment

$$m \xrightarrow{\text{Enc}} c \xrightarrow{f} c^* \xrightarrow{\text{Dec}} m^* \qquad \textit{(Real)}$$

$$\approx \qquad m \xrightarrow{\qquad\quad g \qquad\quad} g(m) \qquad \textit{(Ideal)}$$

- Consider a tamperable communication channel.

- To protect, send $c = \text{Enc}(m)$ along the channel.

- The tampered codeword decodes to some m^*.

- Hope: m^* "looks like" $g(m)$ for some "good" g that we can "tolerate".

We want

- Correctness: $\forall m,\ \text{Dec}(\text{Enc}(m)) = m$.
- Simulation: $\forall\, f \in \mathcal{F},\quad \exists\, g \in \mathcal{G},\quad$ where
  - $\mathcal{F}$ is large and realistic against attacks/channels.
  - $\mathcal{G}$ small and "easy to handle".

Example: Error-correcting codes

$$m \xrightarrow{\text{Enc}} c \xrightarrow{f} c^* \xrightarrow{\text{Dec}} m^* \qquad \textit{(Real)}$$

$$\approx \quad m \xrightarrow{\quad g \quad} g(m) \qquad \textit{(Ideal)}$$

$$\boxed{\quad \mathcal{F} \quad} \longrightarrow \boxed{\mathcal{G} \quad \mathrm{Id}(m) = m}$$

- $\mathcal{G} = \{Id\}$ is “easy to handle".

Example: Error-correcting codes

$$m \xrightarrow{\text{Enc}} c \xrightarrow{f} c^* \xrightarrow{\text{Dec}} m^* \qquad \textit{(Real)}$$

$$\approx \quad m \xrightarrow{\qquad g \qquad} g(m) \qquad \textit{(Ideal)}$$

$$\boxed{\begin{array}{r} \mathcal{F} \\ \Delta(c, c^*) <= \rho \end{array}} \longrightarrow \boxed{\begin{array}{r} \mathcal{G} \\ \text{Id}(m) = m \end{array}}$$

- $\mathcal{G} = \{Id\}$ is “easy to handle".
- $\mathcal{F}$ realistic/useful.
- Constructions: Hadamard, Reed-Solomon, Reed-Muller, etc..

Example: Error-detecting codes

$$m \xrightarrow{\text{Enc}} c \xrightarrow{f} c^* \xrightarrow{\text{Dec}} m^* \qquad \textit{(Real)}$$

$$\approx \quad m \xrightarrow{\qquad g \qquad} g(m) \qquad \textit{(Ideal)}$$

$$\boxed{\mathcal{F}} \longrightarrow \boxed{\begin{array}{l} \mathcal{G} \\ \text{Id}(m) = m \\ \perp(m) = \perp \end{array}}$$

Example: Error-detecting codes

$$m \xrightarrow{\text{Enc}} c \xrightarrow{f} c^* \xrightarrow{\text{Dec}} m^* \qquad \textit{(Real)}$$

$$\approx \quad m \xrightarrow{\quad g \quad} g(m) \qquad \textit{(Ideal)}$$

$$\boxed{\mathcal{F}: \quad \Delta(c, c^*) <= 2\rho} \longrightarrow \boxed{\mathcal{G}: \quad \begin{array}{l} \text{Id}(m) = m \\ \perp(m) = \perp \end{array}}$$

Same constructions as those for ECC.

Example: Error-detecting codes

$$m \xrightarrow{\text{Enc}} c \xrightarrow{f} c^* \xrightarrow{\text{Dec}} m^* \qquad \textit{(Real)}$$

$$\approx \quad m \xrightarrow{\qquad g \qquad} g(m) \qquad \textit{(Ideal)}$$

$$\boxed{\mathcal{F}: \quad f_\delta(c) = c + \delta} \longrightarrow \boxed{\mathcal{G}: \quad \begin{array}{l} \mathrm{Id}(m) = m \\ \perp(m) = \perp \end{array}}$$

AMD Codes: Application in robust fuzzy extractors and secret sharing [C**D**FPW12], NM-codes [DPW10], etc.

Error-correction/detection impossible

$$m \xrightarrow{\mathrm{Enc}} c \xrightarrow{f} c^* \xrightarrow{\mathrm{Dec}} m^* \qquad \textit{(Real)}$$

$$\approx \quad m \xrightarrow{\qquad\qquad g \qquad\qquad} g(m) \qquad \textit{(Ideal)}$$

$$\boxed{\begin{array}{l} \hfill \mathcal{F} \\ \text{Constant} \\ \text{functions} \\ f_{c^*}(c) = c^* \end{array}} \xrightarrow{??} \boxed{\begin{array}{l} \hfill \mathcal{G} \\ \mathrm{Id}(m) = m \\ \perp(m) = \perp \end{array}}$$

Error-correction/detection impossible

$$m \xrightarrow{\text{Enc}} c \xrightarrow{f} c^* \xrightarrow{\text{Dec}} m^* \qquad \textit{(Real)}$$

$$\approx \quad m \xrightarrow{\qquad g \qquad} g(m) \qquad \textit{(Ideal)}$$

$\mathcal{F}$: Constant functions

$f_{c^*}(c) = c^*$

$\not\longrightarrow$

$\mathcal{G}$:

$\text{Id}(m) = m$

$\perp(m) = \perp$

Let $c^* = \text{Enc}(m')$ for some fixed m'.

Thus, $\text{Dec}(c^*) = m' \notin \{m, \perp\}$.

Non-malleable codes

$$m \xrightarrow{\text{Enc}} c \xrightarrow{f} c^* \xrightarrow{\text{Dec}} m^* \qquad \textit{(Real)}$$

$$\approx \quad m \xrightarrow{\quad g \quad} g(m) \qquad \textit{(Ideal)}$$

$$\boxed{\mathcal{F}} \longrightarrow \boxed{\begin{array}{l} \text{NM} \\ \text{Id}(m) = m \\ g_{m^*}(m) = m^* \end{array}}$$

Non-malleable codes

$$m \xrightarrow{\text{Enc}} c \xrightarrow{f} c^* \xrightarrow{\text{Dec}} m^* \qquad \textit{(Real)}$$

$$\approx \quad m \xrightarrow{\qquad g \qquad} g(m) \qquad \textit{(Ideal)}$$

$$\boxed{\mathcal{F}} \longrightarrow \boxed{\begin{array}{l} \text{NM} \\ \text{Id}(m) = m \\ g_{m^*}(m) = m^* \end{array}}$$

Is NM "realistic/easy-to-handle"? When is it useful?

Application of Non-malleable codes

- Consider $\mathrm{Sign}_{sk}(\mathrm{userID}, m)$.
- Task: How to protect sk against tampering attack.
- Encode sk using non-malleable code.
- Thus, $sk^* = \mathrm{Dec}(f(\mathrm{Enc}(sk)))$ is either equal to sk or unrelated.
- Thus, cannot use $\mathrm{Sign}_{sk^*}(\mathrm{userID}, \cdot)$ to forge $\mathrm{Sign}_{sk}(\mathrm{userID}', \cdot)$.

Non-malleable codes: Formal Definition

Let (Enc, Dec) be a coding scheme with Enc **randomized**, and Dec deterministic, s.t. $\forall m \; \mathrm{Dec}(\mathrm{Enc}(m)) = m$,

$$m \xrightarrow{\text{Enc}} c \xrightarrow{f} c^* \xrightarrow{\text{Dec}} m^* \qquad \textit{(Real)}$$

The coding scheme is **non-malleable** w.r.t. family $\mathcal{F}$, if

$\forall f \in \mathcal{F}$,

Non-malleable codes: Formal Definition

Let (Enc, Dec) be a coding scheme with Enc **randomized**, and Dec deterministic, s.t. $\forall m$ Dec(Enc(m)) $=$ m,

$$\mathrm{m} \xrightarrow{\mathrm{Enc}} \mathrm{c} \xrightarrow{\mathrm{f}} \mathrm{c}^* \xrightarrow{\mathrm{Dec}} \mathrm{m}^* \qquad \textit{(Real)}$$

The coding scheme is **non-malleable** w.r.t. family $\mathcal{F}$, if

$\forall f \in \mathcal{F}$, $\exists T$ which is a **probabilistic combination** of:

- constant functions
- identity function

s.t.

Non-malleable codes: Formal Definition

Let (Enc, Dec) be a coding scheme with Enc **randomized**, and Dec deterministic, s.t. $\forall m \ \text{Dec}(\text{Enc}(m)) = m$,

$$m \xrightarrow{\text{Enc}} c \xrightarrow{f} c^* \xrightarrow{\text{Dec}} m^* \qquad (\mathcal{R}eal)$$

The coding scheme is **non-malleable** w.r.t. family $\mathcal{F}$, if

$\forall f \in \mathcal{F}, \ \exists T$ which is a **probabilistic combination** of:

- constant functions
- identity function

s.t.

$$\forall m \in \mathcal{M}, \ m^* \approx T(m) .$$

Non-malleable codes: Formal Definition

Let (Enc, Dec) be a coding scheme with Enc **randomized**, and Dec deterministic, s.t. $\forall m \ \mathrm{Dec}(\mathrm{Enc}(m)) = m$,

$$m \xrightarrow{\text{Enc}} c \xrightarrow{f} c^* \xrightarrow{\text{Dec}} m^* \qquad (\mathcal{Real})$$

The coding scheme is **non-malleable** w.r.t. family $\mathcal{F}$, if

$\forall f \in \mathcal{F}, \ \exists T$ which is a **probabilistic combination** of:

- constant functions
- identity function

s.t.

$$\forall m \in \mathcal{M}, \ m^* \approx T(m) .$$

Note: T is independent of m.
Thus, intuitively, either $m^* = m$ or they are unrelated.

Which realistic families $\mathcal{F}$ can we tolerate?

$$m \xrightarrow{\text{Enc}} c \xrightarrow{f} c^* \xrightarrow{\text{Dec}} m^* \qquad \textit{(Real)}$$

$$\approx \qquad m \xrightarrow{\quad g \quad} g(m) \qquad \textit{(Ideal)}$$

$$\boxed{\mathcal{F}_{\text{all}}} \quad \not\longrightarrow \quad \boxed{\begin{array}{l} \text{NM} \\ \text{Id}(m) = m \\ g_{m^*}(m) = m^* \end{array}}$$

Impossible [DPW10].

Which realistic families $\mathcal{F}$ can we tolerate?

$$m \xrightarrow{\text{Enc}} c \xrightarrow{f} c^* \xrightarrow{\text{Dec}} m^* \qquad \textit{(Real)}$$

$$\approx \qquad m \xrightarrow{\quad\quad g \quad\quad} g(m) \qquad \textit{(Ideal)}$$

$$\boxed{\mathcal{F}_{\text{all}}} \longrightarrow \boxed{\mathcal{F}_{\text{all}}}$$

Impossible [DPW10].

$\forall \; g \in \mathcal{F}_{\text{all}}, \;$ let $\; f(c) = \text{Enc}(g(\text{Dec}(c)))$.

Non-malleable Codes in the t-split-state model

- Tamper t different memory-parts independently

Non-malleable Codes in the t-split-state model

- Tamper t different memory-parts independently
- Application to non-malleable secret-sharing

Non-malleable Codes in the t-split-state model

- Tamper t different memory-parts independently
- Application to non-malleable secret-sharing
- Includes ECC, EDC, Constant functions, bitwise tampering functions but much more

Non-malleable Codes in the t-split-state model

- Tamper t different memory-parts independently
- Application to non-malleable secret-sharing
- Includes ECC, EDC, Constant functions, bitwise tampering functions but much more
- Existential result known [DPW10].

Non-malleable Codes in the t-split-state model

- Tamper t different memory-parts independently
- Application to non-malleable secret-sharing
- Includes ECC, EDC, Constant functions, bitwise tampering functions but much more
- Existential result known [DPW10].
- Efficient construction for family of bitwise-tampering functions ($t = k$, the no. of bits in m)   [DPW10, CG14, FNVW14].

Non-malleable Codes in the t-split-state model

- Tamper t different memory-parts independently
- Application to non-malleable secret-sharing
- Includes ECC, EDC, Constant functions, bitwise tampering functions but much more
- Existential result known [DPW10].
- Efficient construction for family of bitwise-tampering functions ($t = k$, the no. of bits in m) [DPW10, CG14, FNVW14].
- Efficient construction for $t = 2$, $k = 1$ [DKO13]

Non-malleable Codes in the t-split-state model

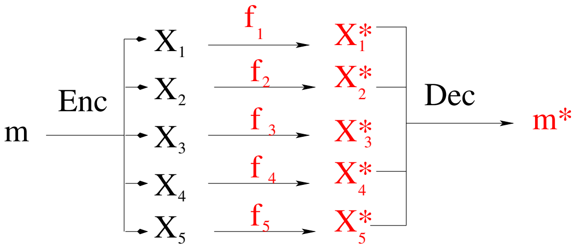
- Tamper t different memory-parts independently
- Application to non-malleable secret-sharing
- Includes ECC, EDC, Constant functions, bitwise tampering functions but much more
- Existential result known [DPW10].
- Efficient construction for family of bitwise-tampering functions ($t = k$, the no. of bits in m) [DPW10, CG14, FNVW14].
- Efficient construction for $t = 2$, $k = 1$ [DKO13]
- Open Question: Efficient construction for t constant, k large.

Non-malleable Codes in the t-split-state model

- Tamper t different memory-parts independently
- Application to non-malleable secret-sharing
- Includes ECC, EDC, Constant functions, bitwise tampering functions but much more
- Existential result known [DPW10].
- Efficient construction for family of bitwise-tampering functions ($t = k$, the no. of bits in m) [DPW10, CG14, FNVW14].
- Efficient construction for $t = 2$, $k = 1$ [DKO13]
- Open Question: Efficient construction for t constant, k large.

  YES (this talk). We show several constructions, including $t = 2$ and constant rate (i.e. code length is $\Theta(k)$).

NM-codes in the t-split state model



The coding scheme is **non-malleable** w.r.t. family $\mathcal{F}_{\text{t-split}}$, if

$\forall\ f_1, \ldots, f_t, \quad \exists T$ which is a **probabilistic combination** of:

- constant functions
- identity function

s.t.

$$\forall m \in \mathcal{M}, \quad m^* \approx T(m) \ .$$

Common outline for our results: Non-malleable reductions [A**D**KO15]

Non-malleable Reduction: Definition [ADKO15]

Let (Enc, Dec) be a coding scheme with Enc **randomized**, and Dec deterministic, s.t. $\forall m \ \text{Dec}(\text{Enc}(m)) = m$,

Non-malleable Reduction: Definition [A**D**KO15]

Let (Enc, Dec) be a coding scheme with Enc **randomized**, and Dec deterministic, s.t. $\forall m\ \mathrm{Dec}(\mathrm{Enc}(m)) = m$,

$$m \xrightarrow{\mathrm{Enc}} c \xrightarrow{f} c^* \xrightarrow{\mathrm{Dec}} m^* \qquad \textit{(Real)}$$

The scheme is a **non-malleable reduction** from $\mathcal{F}$ to $\mathcal{G}$, denoted as $\mathcal{F} \Rightarrow \mathcal{G}$ if

$\forall f \in \mathcal{F}$,

Non-malleable Reduction: Definition [ADKO15]

Let (Enc, Dec) be a coding scheme with Enc **randomized**, and Dec deterministic, s.t. $\forall m \; \mathrm{Dec}(\mathrm{Enc}(m)) = m$,

$$m \xrightarrow{\text{Enc}} c \xrightarrow{f} c^* \xrightarrow{\text{Dec}} m^* \qquad (\mathcal{Real})$$

The scheme is a **non-malleable reduction** from $\mathcal{F}$ to $\mathcal{G}$, denoted as $\mathcal{F} \Rightarrow \mathcal{G}$ if

$\forall f \in \mathcal{F}, \; \exists G$ which is a **probabilistic combination** of functions in $\mathcal{G}$.

Non-malleable Reduction: Definition [ADKO15]

Let (Enc, Dec) be a coding scheme with Enc **randomized**, and Dec deterministic, s.t. $\forall m \ \mathrm{Dec}(\mathrm{Enc}(m)) = m$,

$$\mathrm{m} \xrightarrow{\mathrm{Enc}} \mathrm{c} \xrightarrow{\mathrm{f}} \mathrm{c}^* \xrightarrow{\mathrm{Dec}} \mathrm{m}^* \qquad \mathit{(Real)}$$

The scheme is a **non-malleable reduction** from $\mathcal{F}$ to $\mathcal{G}$, denoted as $\mathcal{F} \Rightarrow \mathcal{G}$ if

$\forall f \in \mathcal{F}, \ \exists G$ which is a **probabilistic combination** of functions in $\mathcal{G}$.

$$\forall m \in \mathcal{M}, \quad m^* \approx G(m) \ .$$

Non-malleable Reduction: Definition [ADKO15]

Let (Enc, Dec) be a coding scheme with Enc **randomized**, and Dec deterministic, s.t. $\forall m$ Dec(Enc(m)) $=$ m,

$$\mathrm{m} \xrightarrow{\mathrm{Enc}} \mathrm{c} \xrightarrow{\mathrm{f}} \mathrm{c}^* \xrightarrow{\mathrm{Dec}} \mathrm{m}^* \qquad (\mathcal{R}eal)$$

The scheme is a **non-malleable reduction** from $\mathcal{F}$ to $\mathcal{G}$, denoted as $\mathcal{F} \Rightarrow \mathcal{G}$ if

$\forall f \in \mathcal{F}$, $\exists G$ which is a **probabilistic combination** of functions in $\mathcal{G}$.

$$\forall m \in \mathcal{M}, \quad m^* \approx G(m) .$$

An NM-code for $\mathcal{F}$ can be viewed as $\mathcal{F} \Rightarrow$ NM, where NM is the function family comprising of

- constant functions
- identity function

Non-malleable Reduction: Composability

Theorem

For all $\mathcal{F}$, $\mathcal{G}$, $\mathcal{H}$, we have that

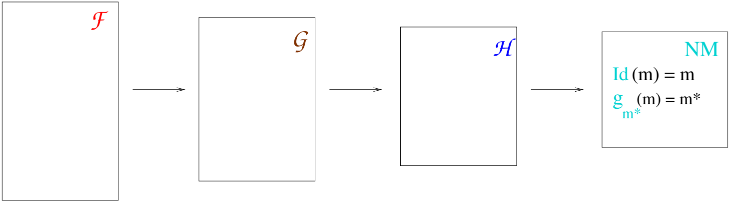
$$\mathcal{F} \Rightarrow \mathcal{G}, \text{ and } \mathcal{G} \Rightarrow \mathcal{H}, \text{ implies } \mathcal{F} \Rightarrow \mathcal{H}\ .$$

Non-malleable Reduction: Composability

Theorem

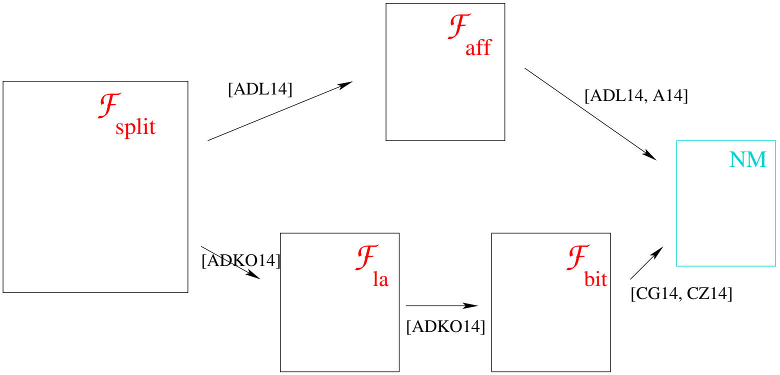
For all $\mathcal{F}$, $\mathcal{G}$, $\mathcal{H}$, we have that

$$\mathcal{F} \Rightarrow \mathcal{G}, \quad \textit{and} \quad \mathcal{G} \Rightarrow \mathcal{H}, \quad \textit{implies} \quad \mathcal{F} \Rightarrow \mathcal{H}\ .$$



Make families simpler, until non-malleable.

Our results



ADL14 gives a scheme for encoding k-bit messages to $\Theta(k^7)$-bit codewords.

ADKO15 gives a scheme for encoding k-bit messages to $\Theta(k)$-bit codewords.

Two simplifying assumptions for the talk

- Will only describe the decoding procedure.

Two simplifying assumptions for the talk

- Will only describe the decoding procedure.
  - $\mathrm{Enc}(m)$ is a random c such that $\mathrm{Dec}(c) = m$.

Two simplifying assumptions for the talk

- Will only describe the decoding procedure.
  - $\mathrm{Enc}(m)$ is a random c such that $\mathrm{Dec}(c) = m$.
  - Subtlety: Enc might be inefficient.

Two simplifying assumptions for the talk

- Will only describe the decoding procedure.
  - $\mathrm{Enc}(m)$ is a random c such that $\mathrm{Dec}(c) = m$.
  - Subtlety: Enc might be inefficient.
  - This can be a problem at times, but for our constructions, we can get around it.

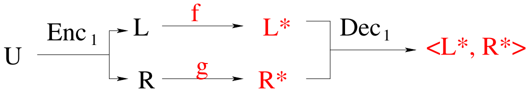
Two simplifying assumptions for the talk

- Will only describe the decoding procedure.
  - $\text{Enc}(m)$ is a random c such that $\text{Dec}(c) = m$.
  - Subtlety: Enc might be inefficient.
  - This can be a problem at times, but for our constructions, we can get around it.
- Argue non-malleability only for a uniformly random message M.

$\mathcal{F}_{\text{split}} \Rightarrow \mathcal{F}_{\text{affine}}$

$U = U_{\mathbb{F}_p}$, $p = \text{poly}(k)$ is a prime

$\text{Enc}_1(U) = L, R \in \mathbb{F}_p^n$ s.t. $\langle L, R \rangle = U$, $n = \text{poly}(\log k)$.



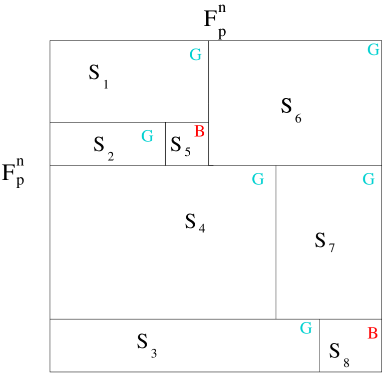
We show:

$$\forall \ f, g, \quad (\langle L, R \rangle, \ \langle f(L), g(R) \rangle) \ \approx \ (U, \ A_{f,g} U + B_{f,g}) \ .$$

Proof Step 1: Partitioning Lemma

Fix f, g. Let $\phi(L, R) := (\langle L, R\rangle,\ \langle f(L), g(R)\rangle)$

$$\mathcal{D} := \{D : D \text{ is a conv. comb. of } (U, aU + b),\ a, b \in \mathbb{F}\}$$



It is enough to partition $\mathbb{F}_p^n \times \mathbb{F}_p^n$ into "good" and "bad" rectangles such that

- If S is a good set, then $\phi(L, R)|_{(L,R)\in S}$ is close to some distribution in $\mathcal{D}$.
- The union of all bad sets has size much smaller than p^{2n}.

Our partitioning

We partition $\mathbb{F}_p^n \times \mathbb{F}_p^n$ into four type of rectangles.

- **Type 1**: $g(R) = a$ for some $a \in \mathbb{F}_p^n$. Then $\phi = (\langle L, R\rangle \ , \ \langle f(L), g(R)\rangle)$ is close to $(U_{\mathbb{F}_p}, \ \langle f(L), a\rangle)$ which belongs to $\mathcal{D}$.

Our partitioning

We partition $\mathbb{F}_p^n \times \mathbb{F}_p^n$ into four type of rectangles.

- **Type 1**: $g(R) = a$ for some $a \in \mathbb{F}_p^n$. Then $\phi = (\langle L, R\rangle\ ,\ \langle f(L), g(R)\rangle)$ is close to $(U_{\mathbb{F}_p},\ \langle f(L), a\rangle)$ which belongs to $\mathcal{D}$.
- **Type 2**: $\phi = (\langle L, R\rangle\ ,\ \langle f(L), g(R)\rangle)$ is close to $U_{\mathbb{F}_p^2}$, which belongs to $\mathcal{D}$.

Our partitioning

We partition $\mathbb{F}_p^n \times \mathbb{F}_p^n$ into four type of rectangles.

- **Type 1**: $g(R) = a$ for some $a \in \mathbb{F}_p^n$. Then $\phi = (\langle L, R\rangle\ ,\ \langle f(L), g(R)\rangle)$ is close to $(U_{\mathbb{F}_p},\ \langle f(L), a\rangle)$ which belongs to $\mathcal{D}$.

- **Type 2**: $\phi = (\langle L, R\rangle\ ,\ \langle f(L), g(R)\rangle)$ is close to $U_{\mathbb{F}_p^2}$, which belongs to $\mathcal{D}$.

- **Type 3**: $f(L) = AL$ for some $A \in \mathbb{F}_p^{n\times n}$, and $A^T g(R) = cR + d$, for $c \in \mathbb{F}_p$, and $d \in \mathbb{F}_p^n$, which implies

$$\phi = (\langle L, R\rangle\ ,\ \ c\langle L, R\rangle + \langle L, d\rangle)\ ,$$

which is in $\mathcal{D}$ if the partition S is large enough.

Our partitioning

We partition $\mathbb{F}_p^n \times \mathbb{F}_p^n$ into four type of rectangles.

- **Type 1**: $g(R) = a$ for some $a \in \mathbb{F}_p^n$. Then $\phi = (\langle L, R\rangle\ ,\ \langle f(L), g(R)\rangle)$ is close to $(U_{\mathbb{F}_p},\ \langle f(L), a\rangle)$ which belongs to $\mathcal{D}$.

- **Type 2**: $\phi = (\langle L, R\rangle\ ,\ \langle f(L), g(R)\rangle)$ is close to $U_{\mathbb{F}_p^2}$, which belongs to $\mathcal{D}$.

- **Type 3**: $f(L) = AL$ for some $A \in \mathbb{F}_p^{n\times n}$, and $A^T g(R) = cR + d$, for $c \in \mathbb{F}_p$, and $d \in \mathbb{F}_p^n$, which implies

$$\phi = (\langle L, R\rangle\ ,\ \ c\langle L, R\rangle + \langle L, d\rangle)\ ,$$

which is in $\mathcal{D}$ if the partition S is large enough.

- **Type 4**: Bad sets.

Our partitioning

We partition $\mathbb{F}_p^n \times \mathbb{F}_p^n$ into four type of rectangles.

- **Type 1**: $g(R) = a$ for some $a \in \mathbb{F}_p^n$. Then $\phi = (\langle L, R\rangle\ ,\ \langle f(L), g(R)\rangle)$ is close to $(U_{\mathbb{F}_p},\ \langle f(L), a\rangle)$ which belongs to $\mathcal{D}$.

- **Type 2**: $\phi = (\langle L, R\rangle\ ,\ \langle f(L), g(R)\rangle)$ is close to $U_{\mathbb{F}_p^2}$, which belongs to $\mathcal{D}$.

- **Type 3**: $f(L) = AL$ for some $A \in \mathbb{F}_p^{n\times n}$, and $A^T g(R) = cR + d$, for $c \in \mathbb{F}_p$, and $d \in \mathbb{F}_p^n$, which implies

$$\phi = (\langle L, R\rangle\ ,\ \ c\langle L, R\rangle + \langle L, d\rangle)\ ,$$

which is in $\mathcal{D}$ if the partition S is large enough.

- **Type 4**: Bad sets.

We show that the set $\mathbb{F}_p^n \times \mathbb{F}_p^n$ can be partitioned into sets of the above four types such that the total size of "bad" sets is much smaller than p^{2n}.

Main tools used for the proof

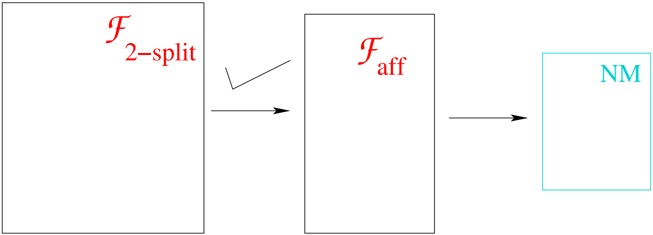
- Linearity test [BSG94, Sam07, San12] : For $f : \mathbb{F}_p^n \mapsto \mathbb{F}_p^n$

  $$\Pr(f(L) - f(L') = f(L - L')) \geq \varepsilon \;\; \Rightarrow \;\; \exists A \;\; \Pr(f(L) = AL) \geq p^{-\log^6(1/\varepsilon)} .$$

  - We need a generalized version, for which we show that essentially the same proof works.

- Hadamard Extractor: $\langle \cdot, \cdot \rangle$ is a strong 2-source extractor.

- (Generalized) Vazirani's XOR Lemma:
  (X_1, X_2) is close to uniform in $\mathbb{F}_p \times \mathbb{F}_p$ if and only if $aX_1 + bX_2$ is close to uniform in $\mathbb{F}_p$ for all $a, b \in \mathbb{F}_p$, not both zero.


$\mathcal{F}_{2\text{-split}}$
$\mathcal{F}_{\mathrm{aff}}$
NM

Step two: $\mathcal{F}_{\text{affine}} \Rightarrow$ NM

$$m \xrightarrow{\text{Enc}_2} c \xrightarrow{h_{A,B}} Ac + B \xrightarrow{\text{Dec}_2} m^*$$

Step two: $\mathcal{F}_{\text{affine}} \Rightarrow$ NM

$$m \xrightarrow{\text{Enc}_2} c \xrightarrow{h_{A,B}} Ac + B \xrightarrow{\text{Dec}_2} m^*$$

Define an *affine-evasive set* $\mathcal{C}$ of $\mathbb{F}_p$ as a set s.t. for C chosen uniformly at random from $\mathcal{C}$,

$\forall\ a,\ b \in \mathbb{F}_p \times \mathbb{F}_p$ s.t. $a \neq 0$ and $(a, b) \neq (1, 0)$

Step two: $\mathcal{F}_{\text{affine}} \Rightarrow$ NM

$$m \xrightarrow{\text{Enc}_2} c \xrightarrow{h_{A,B}} Ac + B \xrightarrow{\text{Dec}_2} m^*$$

Define an *affine-evasive set* $\mathcal{C}$ of $\mathbb{F}_p$ as a set s.t. for C chosen uniformly at random from $\mathcal{C}$,

$\forall\, a,\ b \in \mathbb{F}_p \times \mathbb{F}_p$ s.t. $a \neq 0$ and $(a, b) \neq (1, 0)$

$$\Pr(a \cdot C + b \in \mathcal{C}) \approx 0\ ,$$

Partition $\mathcal{C}$ into equal parts $\mathcal{C}_1, \ldots, \mathcal{C}_{|\mathcal{M}|}$ and define

$$\text{Dec}_2(c) = m, \text{ if } c \in \mathcal{C}_m, \text{ and } \perp, \text{ otherwise}\ .$$

Step two: $\mathcal{F}_{\text{affine}} \Rightarrow$ NM

$$m \xrightarrow{\text{Enc}_2} c \xrightarrow{h_{A,B}} Ac + B \xrightarrow{\text{Dec}_2} m^*$$

Define an *affine-evasive set* $\mathcal{C}$ of $\mathbb{F}_p$ as a set s.t. for C chosen uniformly at random from $\mathcal{C}$,

$\forall\ a,\ b \in \mathbb{F}_p \times \mathbb{F}_p$ s.t. $a \neq 0$ and $(a, b) \neq (1, 0)$

$$\Pr(a \cdot C + b \in \mathcal{C}) \approx 0 \ ,$$

Partition $\mathcal{C}$ into equal parts $\mathcal{C}_1, \ldots, \mathcal{C}_{|\mathcal{M}|}$ and define

$$\text{Dec}_2(c) = m, \text{ if } c \in \mathcal{C}_m, \text{ and } \perp, \text{ otherwise } .$$

Thus,

$$\forall m \in \mathcal{M}, \ \ m^* \approx T(m) \ .$$

Step two: $\mathcal{F}_{\text{affine}} \Rightarrow$ NM

$$m \xrightarrow{\text{Enc}_2} c \xrightarrow{h_{A,B}} Ac + B \xrightarrow{\text{Dec}_2} m^*$$

Define an *affine-evasive set* $\mathcal{C}$ of $\mathbb{F}_p$ as a set s.t. for C chosen uniformly at random from $\mathcal{C}$,

$\forall\ a,\ b \in \mathbb{F}_p \times \mathbb{F}_p$ s.t. $a \neq 0$ and $(a, b) \neq (1, 0)$

$$\Pr(a \cdot C + b \in \mathcal{C}) \approx 0\ ,$$

Partition $\mathcal{C}$ into equal parts $\mathcal{C}_1, \ldots, \mathcal{C}_{|\mathcal{M}|}$ and define

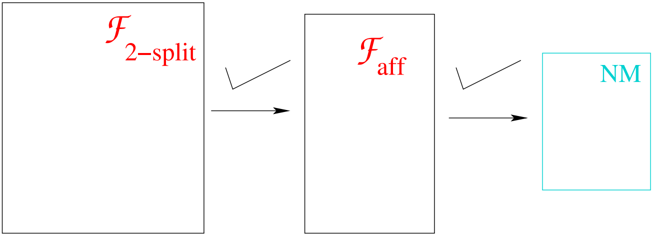
$$\text{Dec}_2(c) = m, \text{ if } c \in \mathcal{C}_m, \text{ and } \perp, \text{ otherwise}\ .$$

Thus,

$$\forall m \in \mathcal{M},\ \ m^* \approx T(m)\ .$$

An affine-evasive set construction modulo p [A14]:

$$S := \left\{ \frac{1}{q} \pmod{p} \;\middle|\; q \text{ is prime},\ \ q < \frac{p^{1/4}}{2} \right\}\ .$$

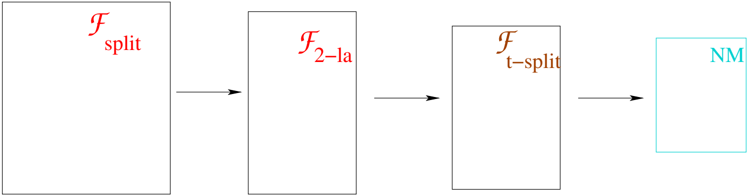


$\mathcal{F}_{2\text{-split}}$
$\mathcal{F}_{\text{aff}}$
NM

Our second result [ADKO15]

NM-reduction from 2-split to t-split for large constant t

k-bit messages $\Longrightarrow \Theta(k)$-bit codewords.



Some natural tampering families

- $\mathcal{S}_n^t$ denotes the tampering family in the *t-split-state model* with each part having length n.

Some natural tampering families

- $\mathcal{S}_n^t$ denotes the tampering family in the *t-split-state model* with each part having length n.
- $\mathcal{L}_n^{\leftarrow t}$ denotes the class of *lookahead manipulation functions* l that can be rewritten as $l = (l_1, \ldots, l_t)$, for $l_i : \{0,1\}^{in} \to \{0,1\}^n$, where

$$l(x) = l_1(x_1)||l_2(x_1, x_2)|| \ldots ||l_i(x_1, \ldots, x_i)|| \ldots ||l_t(x_1, \ldots, x_t)$$

.

$\mathcal{S}^2_{3tn}$ $(\Rightarrow)$ $\mathcal{L}_n^{\leftarrow t}$

Quentin: Q, S_1		Wendy W
S_1	$\xrightarrow{S_1}$	
	$\xleftarrow{R_1}$	$R_1 = \mathrm{Ext}(W; S_1)$
$S_2 = \mathrm{Ext}(Q; R_1)$	$\xrightarrow{S_2}$	
	$\xleftarrow{R_2}$	$R_2 = \mathrm{Ext}(W; S_2)$
	$\ldots$	
$S_t = \mathrm{Ext}(Q; R_{t-1})$	$\xrightarrow{S_t}$	
		$R_t = \mathrm{Ext}(W; S_t)$

Figure: Alternating Extraction

$\mathcal{S}^2_{3tn}$ $(\Rightarrow)$ $\mathcal{L}_n^{\leftarrow t}$

Quentin: Q, S_1		Wendy W
S_1	$\xrightarrow{S_1}$	
	$\xleftarrow{R_1}$	$R_1 = \mathrm{Ext}(W; S_1)$
$S_2 = \mathrm{Ext}(Q; R_1)$	$\xrightarrow{S_2}$	

- $\mathrm{Dec}((Q, S_1), W) = S_1, \ldots, S_t$.
- Alternating Extraction Theorem [DP07] shows:

$$S_{i+1}, \ldots, S_t \approx U, \text{ given } S_1, \ldots, S_i, S'_1, \ldots, S'_i \; .$$

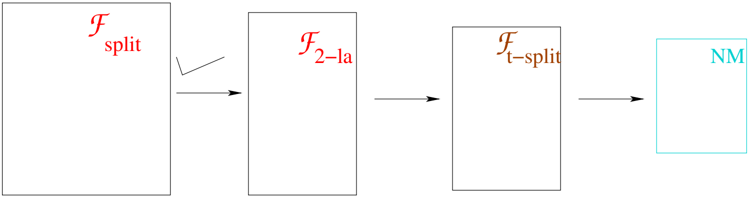
- Intuitively, this implies

$$\forall i, \quad S'_i \text{ is independent of } S_{i+1}, \ldots, S_t \; .$$

$\mathcal{S}^2_{3tn}$ $(\Rightarrow)$ $\mathcal{L}_n^{\leftarrow t}$

Quentin: Q, S_1		Wendy W
S_1	$\xrightarrow{S_1}$	
	$\xleftarrow{R_1}$	$R_1 = \mathrm{Ext}(W; S_1)$
$S_2 = \mathrm{Ext}(Q; R_1)$	$\xrightarrow{S_2}$	
	$\xleftarrow{R_2}$	$R_2 = \mathrm{Ext}(W; S_2)$
	$\dots$	
$S_t = \mathrm{Ext}(Q; R_{t-1})$	$\xrightarrow{S_t}$	
		$R_t = \mathrm{Ext}(W; S_t)$

Figure: Alternating Extraction


$\mathcal{F}_{\text{split}}$
$\mathcal{F}_{2\text{-la}}$
$\mathcal{F}_{\text{t-split}}$
NM

$\mathcal{L}_{2t\ell}^{\leftarrow t} \times \mathcal{L}_{2t\ell}^{\leftarrow t} \Rightarrow \mathcal{S}_{\ell}^{t}$

Define the reduction by the following:

$$\mathrm{Dec}(L, R) := (\langle L_t, R_1\rangle, \langle L_{t-1}, R_2\rangle, \ldots \langle L_1, R_t\rangle) \ ,$$

where $\langle\cdot,\cdot\rangle$ is the ℓ-bit inner product (interpreting L_i, R_i as elements of $\mathbb{F}_{2^n}^{2t}$.

$\mathcal{L}_{2t\ell}^{\leftarrow t} \times \mathcal{L}_{2t\ell}^{\leftarrow t} \;\Rightarrow\; \mathcal{S}_{\ell}^{t}$

Define the reduction by the following:

$$\mathrm{Dec}(L, R) := (\langle L_t, R_1\rangle, \langle L_{t-1}, R_2\rangle, \ldots \langle L_1, R_t\rangle) \;,$$

where $\langle \cdot, \cdot \rangle$ is the ℓ-bit inner product (interpreting L_i, R_i as elements of $\mathbb{F}_{2^n}^{2t}$.

Intuitively, the result follows from the observation (using the Hadamard two-source extractor property) that $b_i = \langle L_{t-i+1}, R_i\rangle$ is close to uniform given $b'_j = \langle L'_{t-j+1}, R'_j\rangle$ for $j \neq i$.

$\mathcal{L}_{2t\ell}^{\leftarrow t} \times \mathcal{L}_{2t\ell}^{\leftarrow t} \Rightarrow \mathcal{S}_{\ell}^{t}$

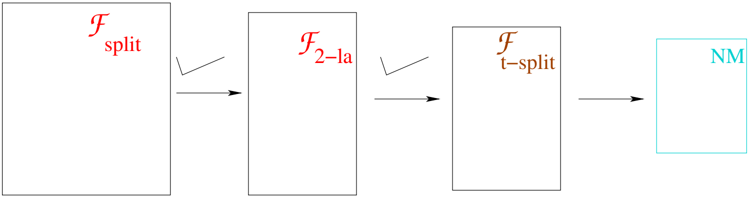
Define the reduction by the following:

$$\mathrm{Dec}(L, R) := (\langle L_t, R_1 \rangle, \langle L_{t-1}, R_2 \rangle, \ldots \langle L_1, R_t \rangle) ,$$

where $\langle \cdot, \cdot \rangle$ is the ℓ-bit inner product (interpreting L_i, R_i as elements of $\mathbb{F}_{2^n}^{2t}$.

Intuitively, the result follows from the observation (using the Hadamard two-source extractor property) that $b_i = \langle L_{t-i+1}, R_i \rangle$ is close to uniform given $b'_j = \langle L'_{t-j+1}, R'_j \rangle$ for $j \neq i$.

Formal proof: More subtle due to joint distributions. See paper.



$\mathcal{F}_{\text{split}}$
$\mathcal{F}_{2\text{-la}}$
$\mathcal{F}_{\text{t-split}}$
NM

Summarizing and Composing the two reductions

We showed:

- $\mathcal{S}^2_{3tn}$ $(\Rightarrow)$ $\mathcal{L}^{\leftarrow t}_{n}$

- $\mathcal{L}^{\leftarrow t}_{2t\ell} \times \mathcal{L}^{\leftarrow t}_{2t\ell} \;\Rightarrow\; \mathcal{S}^{t}_{\ell}$

Summarizing and Composing the two reductions

We showed:

- $\mathcal{S}^2_{3tn} \;\; (\Rightarrow) \;\; \mathcal{L}^{\leftarrow t}_n$
- $\mathcal{L}^{\leftarrow t}_{2t\ell} \times \mathcal{L}^{\leftarrow t}_{2t\ell} \;\; \Rightarrow \;\; \mathcal{S}^t_\ell$

By composing, we get

$$\mathcal{S}^4_{6t^2\ell} \;\; (\Rightarrow) \;\; \mathcal{S}^t_\ell \, .$$

Summarizing and Composing the two reductions

We showed:

- $\mathcal{S}^2_{3tn}$ $(\Rightarrow)$ $\mathcal{L}^{\leftarrow t}_n$

- $\mathcal{L}^{\leftarrow t}_{2t\ell} \times \mathcal{L}^{\leftarrow t}_{2t\ell} \;\Rightarrow\; \mathcal{S}^t_\ell$

By composing, we get

$$\mathcal{S}^4_{6t^2\ell} \;(\Rightarrow)\; \mathcal{S}^t_\ell \,.$$

This, however is not efficiently invertible. We can add a fifth part to make it efficiently invertible.

Summarizing and Composing the two reductions

We showed:

- $\mathcal{S}^2_{3tn}$ $(\Rightarrow)$ $\mathcal{L}^{\leftarrow t}_n$

- $\mathcal{L}^{\leftarrow t}_{2t\ell} \times \mathcal{L}^{\leftarrow t}_{2t\ell} \;\Rightarrow\; \mathcal{S}^t_\ell$

By composing, we get

$$\mathcal{S}^4_{6t^2\ell} \ (\Rightarrow)\ \mathcal{S}^t_\ell\ .$$

This, however is not efficiently invertible. We can add a fifth part to make it efficiently invertible.

Using another more involved construction, we can modify the first reduction to get the following efficiently invertible reduction.

- $\mathcal{S}^2_{O(t^3 n)} \;\Rightarrow\; \mathcal{L}^{\leftarrow t}_n \times \mathcal{L}^{\leftarrow t}_n \cup \ldots$ (only works for constant t) .

Summarizing and Composing the two reductions

We showed:

- $\mathcal{S}^2_{3tn} \ (\Rightarrow) \ \mathcal{L}^{\leftarrow t}_n$
- $\mathcal{L}^{\leftarrow t}_{2t\ell} \times \mathcal{L}^{\leftarrow t}_{2t\ell} \ \Rightarrow \ \mathcal{S}^t_\ell$

By composing, we get

$$\mathcal{S}^4_{6t^2\ell} \ (\Rightarrow) \ \mathcal{S}^t_\ell \, .$$

This, however is not efficiently invertible. We can add a fifth part to make it efficiently invertible.

Using another more involved construction, we can modify the first reduction to get the following efficiently invertible reduction.

- $\mathcal{S}^2_{O(t^3 n)} \ \Rightarrow \ \mathcal{L}^{\leftarrow t}_n \times \mathcal{L}^{\leftarrow t}_n \cup \ldots$ (only works for constant t) .

This implies:

$$\mathcal{S}^2_{\mathrm{poly}(t)\cdot\ell} \ \Rightarrow \ \mathcal{S}^t_\ell \, .$$

Concluding Non-malleability

Our work combined with an independent work [CZ14] gives constant rate 2-split NM-Codes.

Concluding Non-malleability

Our work combined with an independent work [CZ14] gives constant rate 2-split NM-Codes.

[CZ14] showed: $\mathcal{S}^{10}_{\Theta(\ell)} \quad \Rightarrow \quad \mathrm{NM}_{\ell}$.

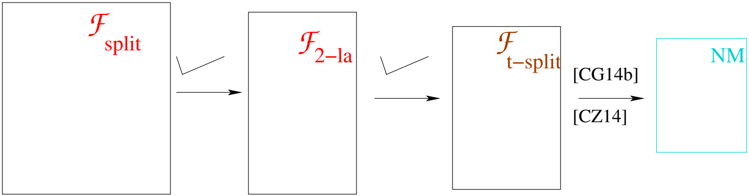
Concluding Non-malleability

Our work combined with an independent work [CZ14] gives constant rate 2-split NM-Codes.

[CZ14] showed: $\mathcal{S}^{10}_{\Theta(\ell)} \Rightarrow \mathsf{NM}_\ell$.

This combined with our reduction gives:

$$\mathcal{S}^{2}_{\Theta(\ell)} \Rightarrow \mathsf{NM}_\ell \; .$$



Future work

The following are major open questions in this area.

- Optimizing the rate of the NM-code construction in split-state model, either by improving our proof techniques, or using some other construction.
- Proposing other useful tampering models.
- Other applications of NM-codes. There has been some recent work in this direction by [CMTV14] and [AGMPP14].

Thank You