Nonmalleable Digital Lockers and Robust Fuzzy Extractors in the Plain Model

Daniel Apon [1] Chloe Cachet [2] Benjamin Fuller [2]
Peter Hall [3] Feng-Hao Liu [4]

[1]MITRE [2]UConn [3]NYU [4]FAU

December 9, 2022

Random Oracle Model

Random Oracles: all parties have access to truly random function:

$$H : \{0,1\}^* \to \{0,1\}^w$$

Random Oracle Model

Random Oracles: all parties have access to truly random function:

$$H : \{0,1\}^* \to \{0,1\}^w$$

The problem: It cannot exist in general [CGH04].
roadblock to realizing cryptographic primitives in plain model

Most Common — Use a heuristic hash function.
However, we want provable security.

Random Oracle Model

Random Oracles: all parties have access to truly random function:

$$H : \{0,1\}^* \to \{0,1\}^w$$

The problem: It cannot exist in general [CGH04].
roadblock to realizing cryptographic primitives in plain model

Most Common — Use a heuristic hash function.
However, we want provable security.

Idea

If we isolate certain properties we need for applications, may be able to get provable security by realizing these.

Random Oracle Model

Random Oracles: all parties have access to truly random function:

$$H : \{0,1\}^* \to \{0,1\}^w$$

The problem: It cannot exist in general [CGH04].
roadblock to realizing cryptographic primitives in plain model

Most Common — Use a heuristic hash function.
However, we want provable security.

Idea

If we isolate certain properties we need for applications, may be able to get provable security by realizing these.

In this work, we isolate and realize oracle hashing and nonmalleability.

Point Functions:

$$I_{\mathsf{val}}(\mathsf{val}') = \begin{cases} 1 & \mathsf{val}' = \mathsf{val} \\ 0 & \text{else} \end{cases}$$

Point Functions:

$$I_{\mathsf{val}}(\mathsf{val}') = \begin{cases} 1 & \mathsf{val}' = \mathsf{val} \\ 0 & \text{else} \end{cases}$$

- Hide everything about I_{val} except input/output behavior

$$\text{Obfuscate}: \ \mathcal{O}(\mathsf{val}) = \widetilde{O}$$
$$\text{On use:} \ \widetilde{O}(x) \equiv I_{\mathsf{val}}(x)$$

- **VBB obfuscation**: Ensure the following is negligible

$$|\Pr[\mathcal{A}(\widetilde{O}) = \mathcal{P}(\mathsf{val}) | \widetilde{O} \leftarrow \mathcal{O}(I_{\mathsf{val}})] - \Pr[\mathcal{S}^{I_{\mathsf{val}}(\cdot)}(1^{\lambda}) = \mathcal{P}(\mathsf{val})]|$$

Issue: May be easy to take $\mathcal{O}(\mathsf{val})$ and **obliviously** tamper to some "related" point $\mathsf{val}' = f(\mathsf{val})$. "Preventing this" is called **nonmalleability**.

Nonmalleability and Composition

Issue: May be easy to take $\mathcal{O}(\mathsf{val})$ and **obliviously** tamper to some "related" point $\mathsf{val}' = f(\mathsf{val})$. "Preventing this" is called **nonmalleability**.

Note

"Preventing" mauling to "related" points makes a lot of sense with trusted setup or ROs, but tricky in plain model. More on this later.

Nonmalleability and Composition

Issue: May be easy to take $\mathcal{O}(\mathsf{val})$ and **obliviously** tamper to some "related" point $\mathsf{val}' = f(\mathsf{val})$. "Preventing this" is called **nonmalleability**.

Additionally, want obfuscation for multibit output:

$$I_{\mathsf{val}}(\mathsf{val}') = \begin{cases} 1 & \mathsf{val}' = \mathsf{val} \\ 0 & \mathsf{else} \end{cases} \implies I_{\mathsf{val},\mathsf{key}}(\mathsf{val}') = \begin{cases} \mathsf{key} & \mathsf{val}' = \mathsf{val} \\ \perp & \mathsf{else} \end{cases}$$

Nonmalleability and Composition

Issue: May be easy to take $\mathcal{O}(\mathsf{val})$ and **obliviously** tamper to some "related" point $\mathsf{val}' = f(\mathsf{val})$. "Preventing this" is called **nonmalleability**.

Additionally, want obfuscation for multibit output:

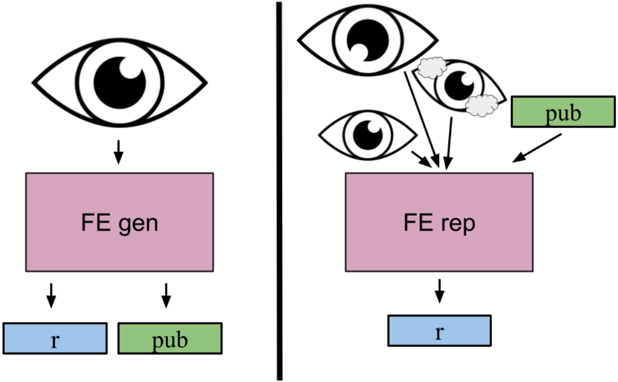
$$I_{\mathsf{val}}(\mathsf{val}') = \begin{cases} 1 & \mathsf{val}' = \mathsf{val} \\ 0 & \text{else} \end{cases} \implies I_{\mathsf{val},\mathsf{key}}(\mathsf{val}') = \begin{cases} \mathsf{key} & \mathsf{val}' = \mathsf{val} \\ \perp & \text{else} \end{cases}$$

- Maybe nonmalleability over both inputs here

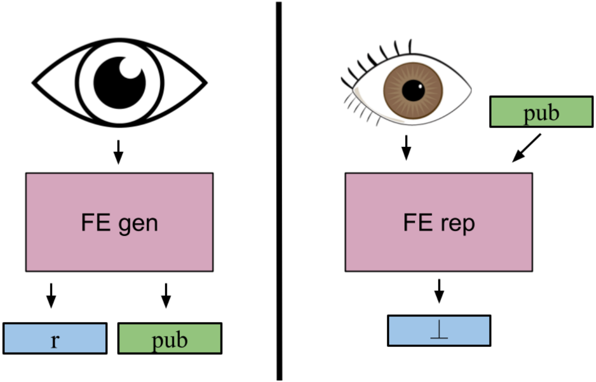
Called a nonmalleable point obfuscation with multibit output...or a **digital locker**

Motivation — (Robust) Fuzzy Extractors

Fuzzy Extractors: retrieve stable random strings from lower entropy and noisy inputs.

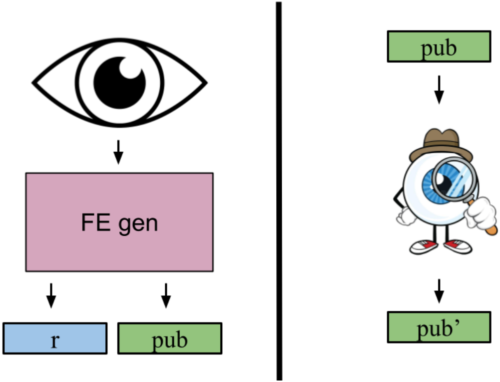


Fuzzy Extractors: retrieve stable random strings from lower entropy and noisy inputs.



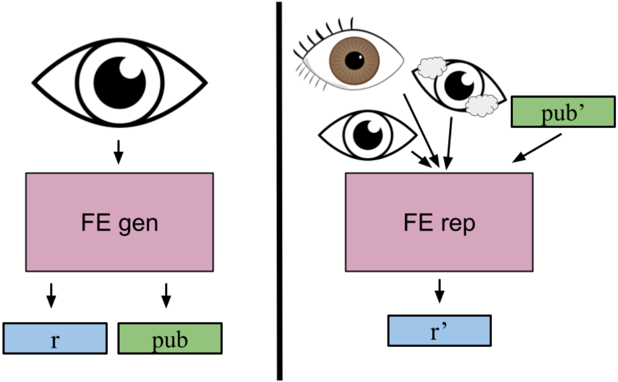
Motivation — (Robust) Fuzzy Extractors

Robustness [Boy04, BDK+05]: Should be hard for adversary with existing *pub* to output *pub*$'$ which reproduces to different $r' \neq r$



Motivation — (Robust) Fuzzy Extractors

Robustness [Boy04, BDK+05]: Should be hard for adversary with existing *pub* to output *pub*$'$ which reproduces to different $r' \neq r$



Motivation — (Robust) Fuzzy Extractors

Robust Fuzzy Extractors over low-entropy inputs are known in ROM and Common Reference String (CRS) model. Meanwhile inputs with entropy less than half their length have been a long-standing barrier in the plain model.

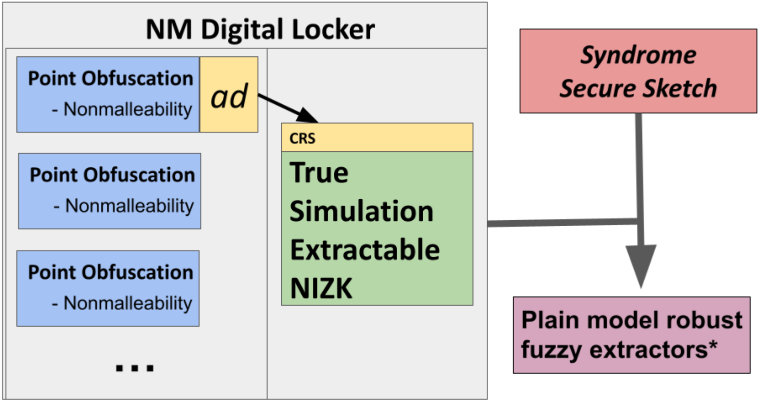
Scheme	Model	Security	SS errors	$H_\infty < 1/2?$
[Boy04],[Boy07]	RO	IT	t	✓
[DKK+12]	Plain	IT	t	X
[CDF+08]	CRS	IT	t	X
[WL18]	CRS	Comp.	$2t$	✓
[FT21]	CRS*	Comp.	$2t$	✓

Motivation — Fuzzy Extractors

Robust Fuzzy Extractors over low-entropy inputs are known in ROM and Common Reference String (CRS) model. Meanwhile inputs with entropy less than half their length have been a long-standing barrier in the plain model.

Scheme	Model	Security	SS errors	$H_\infty < 1/2$?
[Boy04],[Boy07]	RO	IT	t	✓
[DKK+12]	Plain	IT	t	X
[CDF+08]	CRS	IT	t	X
[WL18]	CRS	Comp.	$2t$	✓
[FT21]	CRS*	Comp.	$2t$	✓
This work	Plain	Comp.	$2t$	✓
This work	Plain	Comp.	$2t$	✓

The Plan

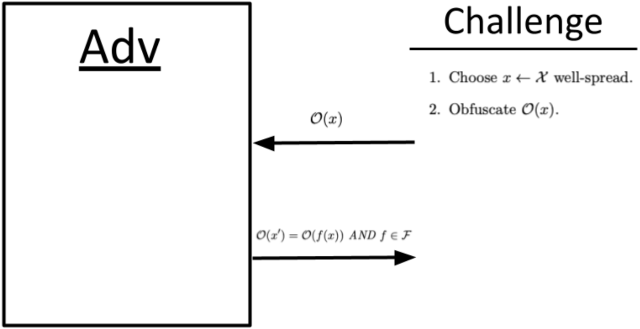


NM Digital Locker
Point Obfuscation
- Nonmalleability
ad
Point Obfuscation
- Nonmalleability
Point Obfuscation
- Nonmalleability
...
CRS
True Simulation Extractable NIZK
Syndrome Secure Sketch
Plain model robust fuzzy extractors*

Nonmalleable Digital Lockers

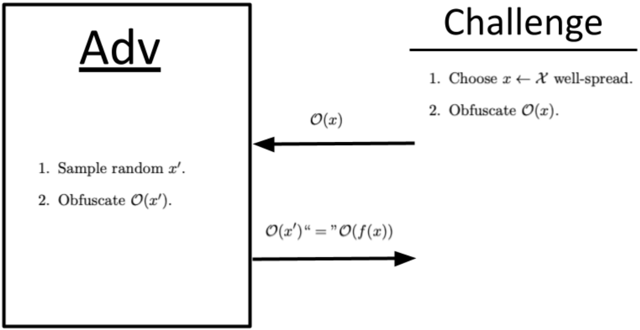
Komargodski and Yogev [KY18]:

- Nonmalleability defined as adversary outputting tampering function from $\mathcal{F}$



Komargodski and Yogev [KY18]:

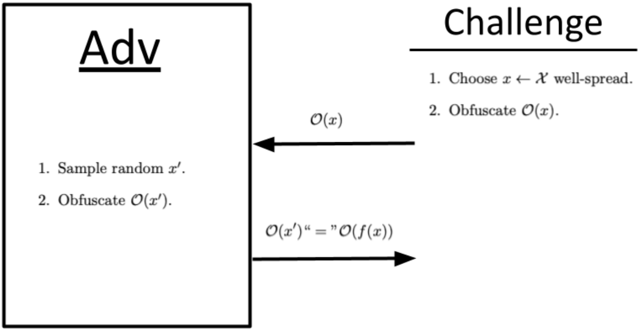
- Nonmalleability defined as adversary outputting tampering function from $\mathcal{F}$



Background — Plain Model NM Point Obfuscation

Komargodski and Yogev [KY18]:

- Nonmalleability defined as adversary outputting tampering function from $\mathcal{F}$



From PO to DL

To get to multibit output, most common method is *Real-or-Random* composition of point obfuscations.

- For each bit of the output key, append $\mathcal{O}(\text{val})$ if the bit is 1 and $\mathcal{O}(r)$ for some random value r if the bit is 0.
- DL functions by reconstructing key bit-by-bit.

From PO to DL

To get to multibit output, most common method is *Real-or-Random* composition of point obfuscations.

- For each bit of the output key, append $\mathcal{O}(\text{val})$ if the bit is 1 and $\mathcal{O}(r)$ for some random value r if the bit is 0.
- DL functions by reconstructing key bit-by-bit.

HOWEVER, this requires...

- Point obfuscations composability
- Some way to protect key.

Previous work [FF20] required a CRS to achieve key nonmalleability.

GOAL: Remove CRS to bring NMDLs into plain model!

Nonmalleable Point Obfuscations with Associated Data

Definition

Let $\rho \in \mathbb{N}, \mathcal{X}$ be a family of distributions, and $\mathcal{F}$ be a family of functions. Then, a $(\mathcal{F}, \mathcal{X}, \rho)$-**Nonmalleable Point Obfuscation with Associated Data** is defined as

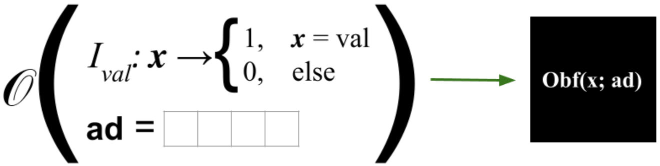
$$\mathsf{lockPoint}(x; ad) := (ad; \mathsf{unlockPoint}(x; ad)),$$

where $x \leftarrow \mathcal{X}, ad \in \{0,1\}^{\rho}$, and unlockPoint satisfies *completeness*, *VBB security*, and *nonmalleability*.

Nonmalleable Point Obfuscations with Associated Data

Definition

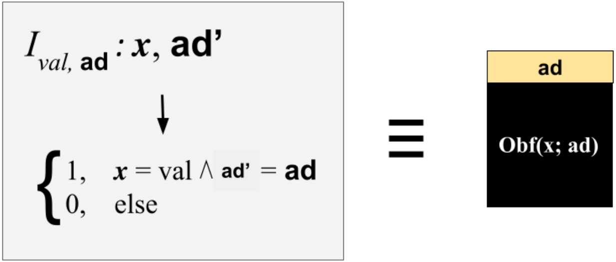
A $(\mathcal{F}, \mathcal{X}, \rho)$-**Nonmalleable Point Obfuscation with Associated Data** is defined as $\mathsf{lockPoint}(x; ad) := (ad; \mathsf{unlockPoint}(x; ad)), \ldots$



Nonmalleable Point Obfuscations with Associated Data

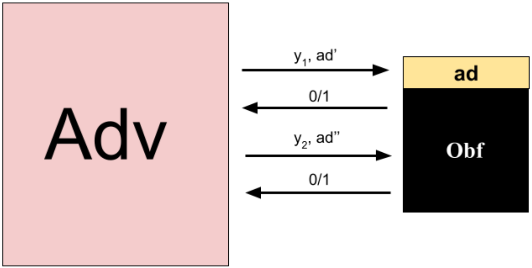
Definition

. . . satisfying *completeness*, . . .



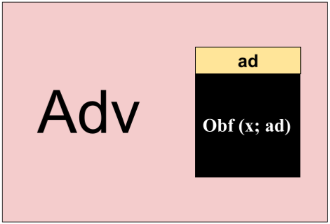
Definition

. . . *VBB security*, . . .



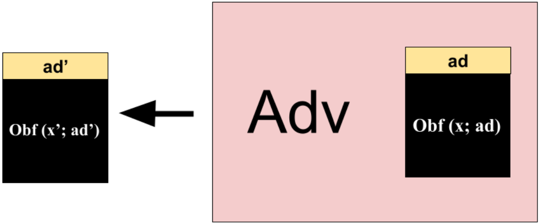
Definition

. . . and *nonmalleability*.



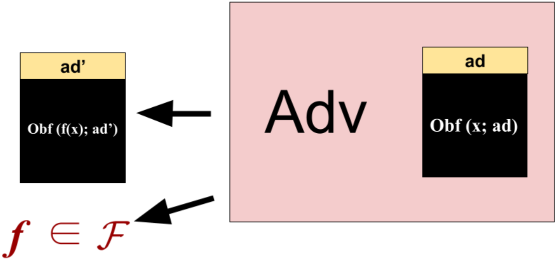
Definition

. . . and *nonmalleability*.



Definition

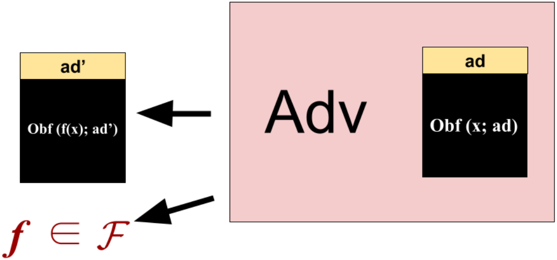
. . . and *nonmalleability*.



Nonmalleable Point Obfuscations with Associated Data

Definition

. . . and *nonmalleability*.



Note

The adversary succeeds if they tamper the ad or underlying point function (or both).

Assumptions

Bartusek, Ma, and Zhandry [BMZ19] studied fixed generator assumptions (toward point obfuscation!) in the GGM, showed following holds there:

Assumption

For $x \leftarrow \mathcal{X}$ well-spread and random r the following is $\mathsf{negl}(\lambda)$ *for all PPT A:*

$$|\Pr[\mathcal{A}(\{k_i, g^{k_i x + x^i}\}_{i \in [2,\tau]}) = 1] - \Pr[\mathcal{A}(\{k_i, g^{k_i r + r^i}\}_{i \in [2,\tau]}) = 1]|.$$

$$\Longrightarrow$$

Assumption

For $x \leftarrow \mathcal{X}$ well-spread, the following is $\mathsf{negl}(\lambda)$ *for all PPT A:*

$$\Pr[g^x \leftarrow \mathcal{A}(\{k_i, g^{k_i x + x^i}\}_{i \in [2,\tau]})].$$

Constructing $NMPO_{ad}$

Constructing $NMPO_{ad}$

1. Sample random values

$$c_1, c_2, c_3, c_4, c_5 \xleftarrow{\$} \mathbb{Z}_{p(\lambda)}$$

Constructing $NMPO_{ad}$

1. Sample random values

$$c_1, c_2, c_3, c_4, c_5 \xleftarrow{\$} \mathbb{Z}_{p(\lambda)}$$

2. Sample $ad \leftarrow \{0,1\}^\rho$ and form

$$p_{1,ad,c_1}(\mathsf{val}) = c_1\mathsf{val} + \sum_{i=1}^{\rho} ad_i \mathsf{val}^{i+1} + \sum_{i=\rho+2}^{\rho+6} \mathsf{val}^i,$$

$$p_{2,c_2}(\mathsf{val}) = c_2\mathsf{val} + \mathsf{val}^{\rho+7},$$

$$p_{3,c_3}(\mathsf{val}) = c_3\mathsf{val} + \mathsf{val}^{\rho+8},$$

$$p_{4,c_4}(\mathsf{val}) = c_4\mathsf{val} + \mathsf{val}^{\rho+9},$$

$$p_{5,c_5}(\mathsf{val}) = c_5\mathsf{val} + \mathsf{val}^{\rho+10}.$$

Constructing $NMPO_{ad}$

1. Sample random values

$$c_1, c_2, c_3, c_4, c_5 \xleftarrow{\$} \mathbb{Z}_{p(\lambda)}$$

2. Sample $ad \leftarrow \{0,1\}^\rho$ and form

$$p_{1,ad,c_1}(\mathsf{val}) = c_1\mathsf{val} + \sum_{i=1}^{\rho} ad_i \mathsf{val}^{i+1} + \sum_{i=\rho+2}^{\rho+6} \mathsf{val}^i,$$

3. Define

$$\mathsf{lockPoint}(\mathsf{val}, ad; c_1, c_2, c_3, c_4, c_5) \stackrel{def}{=} \begin{pmatrix} c_1, & [p_{1,ad,c_1}(\mathsf{val})]_g \\ c_2, & [p_{2,c_2}(\mathsf{val})]_g \\ c_3, & [p_{3,c_3}(\mathsf{val})]_g \\ c_4, & [p_{4,c_4}(\mathsf{val})]_g \\ c_5, & [p_{5,c_5}(\mathsf{val})]_g \end{pmatrix}$$

Note

Reminder: Require nonmalleability for adversaries **outputting** f *and either (1) mauling* x *or (2) mauling* ad *and letting* $f = id$.

Proof route:

Lemma (Lemma 4.3)

Given any degree-ρ *polynomial* P*, no adversary can maul*

$$\mathcal{O}_P(x) = (c_1, [c_1 x + xP(x) + \sum_{i=\rho+2}^{\rho+6} x^i]_g)$$

to any $\mathcal{O}_{P'}(f(x))$ *for any degree-*ρ *polynomial* P' *and* $f \in \mathcal{F}$ *(with non-negligible probability).*

Nonmalleability Proof Roadmap

Note

Reminder: Require nonmalleability for adversaries **outputting** f *and either (1) mauling* x *or (2) mauling* ad *and letting* $f = id$.

Proof route:

Lemma (Lemma 4.5)

Given that x *is not tampered, then for any* $ad \in \{0,1\}^\rho$, *no adversary can maul*

$$\mathcal{O}_{ad}(x) = (c_1, [c_1 x + \sum_{i=1}^{\rho} ad_i x^{i+1} + \sum_{i=\rho+2}^{\rho+6} x^i]_g)$$

to $\mathcal{O}_{ad'}(x)$ *for any* $ad' \neq ad$ *(with non-negligible probability).*

Nonmalleability Proof Roadmap

Note

Reminder: Require nonmalleability for adversaries **outputting** f *and either (1) mauling* x *or (2) mauling* ad *and letting* $f = id$.

Proof route:

- **Lemma 4.3** ensures that any non-identity shifts of x are hard to reach
  - Namely, any $\mathcal{O}(f(x))$ is outside the span of elements in $\mathcal{O}(x)$.
- **Lemma 4.5** ensures any maulings of ad when $f = id$ are hard to reach.

Lemma 4.5

We have $f = \mathrm{id}$ and $ad' \neq ad$. So, adversary is given

$$\mathcal{O}_{ad}(x) = (c_1, [c_1 x + \sum_{i=1}^{\rho} ad_i x^{i+1} + \sum_{i=\rho+2}^{\rho+6} x^i]_g)$$

and must construct

$$\mathcal{O}_{ad'}(x) = (c_1', [c_1' x + \sum_{i=1}^{\rho} ad_i' x^{i+1} + \sum_{i=\rho+2}^{\rho+6} x^i]_g)$$

Lemma 4.5

We have $f = \mathrm{id}$ and $ad' \neq ad$. So, adversary is given

$$\mathcal{O}_{ad}(x) = (c_1, [c_1 x + \sum_{i=1}^{\rho} ad_i x^{i+1} + \sum_{i=\rho+2}^{\rho+6} x^i]_g)$$

and must construct

$$\mathcal{O}_{ad'}(x) = (c_1', [c_1' x + \sum_{i=1}^{\rho} ad_i' x^{i+1} + \sum_{i=\rho+2}^{\rho+6} x^i]_g)$$

$\Longrightarrow$ So, ad' differs from ad one at least one bit $ad_i' \neq ad_i$.

Lemma 4.5

We have $f = \mathrm{id}$ and $ad' \neq ad$. So, adversary is given

$$\mathcal{O}_{ad}(x) = (c_1, [c_1 x + \sum_{i=1}^{\rho} ad_i x^{i+1} + \sum_{i=\rho+2}^{\rho+6} x^i]_g)$$

and must construct

$$\mathcal{O}_{ad'}(x) = (c_1', [c_1' x + \sum_{i=1}^{\rho} ad_i' x^{i+1} + \sum_{i=\rho+2}^{\rho+6} x^i]_g)$$

$\implies$ So, ad' differs from ad one at least one bit $ad_i' \neq ad_i$.

If $ad_i' = 1$ and $ad_i = 0$, then adversary's linear term (c_1') must coincide with term from assumption

$$k_i, g^{k_i x + x^i}.$$

However, never given any input related to k_i

Lemma 4.5

We have $f = \mathrm{id}$ and $ad' \neq ad$. So, adversary is given

$$\mathcal{O}_{ad}(x) = (c_1, [c_1 x + \sum_{i=1}^{\rho} ad_i x^{i+1} + \sum_{i=\rho+2}^{\rho+6} x^i]_g)$$

and must construct

$$\mathcal{O}_{ad'}(x) = (c_1', [c_1' x + \sum_{i=1}^{\rho} ad_i' x^{i+1} + \sum_{i=\rho+2}^{\rho+6} x^i]_g)$$

$\implies$ So, ad' differs from ad one at least one bit $ad_i' \neq ad_i$.

If $ad_i' = 0$ and $ad_i = 1$, then a sort of inverse is true — adversary extracted k_i from the other terms to remove it from input c_1.

Lemma 4.5

We have $f = \text{id}$ and $ad' \neq ad$. So, adversary is given

$$\mathcal{O}_{ad}(x) = (c_1, [c_1 x + \sum_{i=1}^{\rho} ad_i x^{i+1} + \sum_{i=\rho+2}^{\rho+6} x^i]_g)$$

and must construct

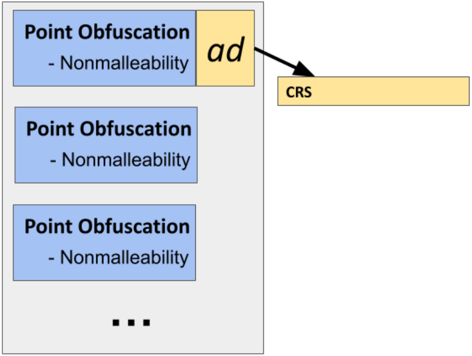
$$\mathcal{O}_{ad'}(x) = (c_1', [c_1' x + \sum_{i=1}^{\rho} ad_i' x^{i+1} + \sum_{i=\rho+2}^{\rho+6} x^i]_g)$$

$\implies$ So, ad' differs from ad one at least one bit $ad_i' \neq ad_i$.

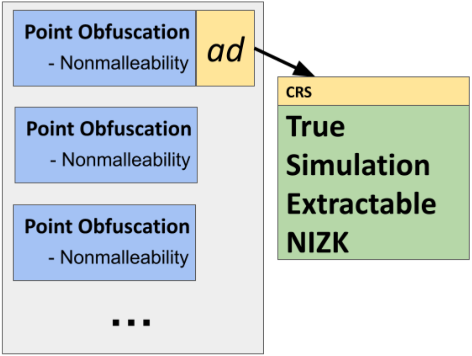
If $ad_i' = 0$ and $ad_i = 1$, then a sort of inverse is true — adversary extracted k_i from the other terms to remove it from input c_1.

$\implies$ In either case, their success probability is small.

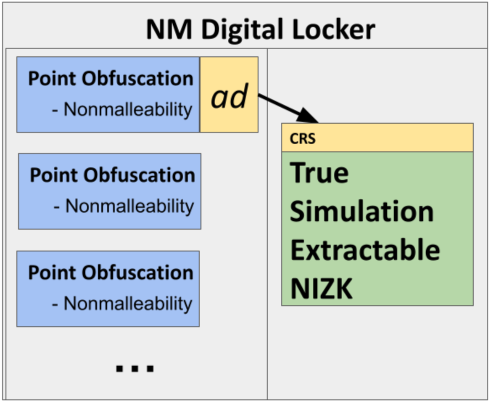
Constructing Nonmalleable Digital Lockers



Constructing Nonmalleable Digital Lockers

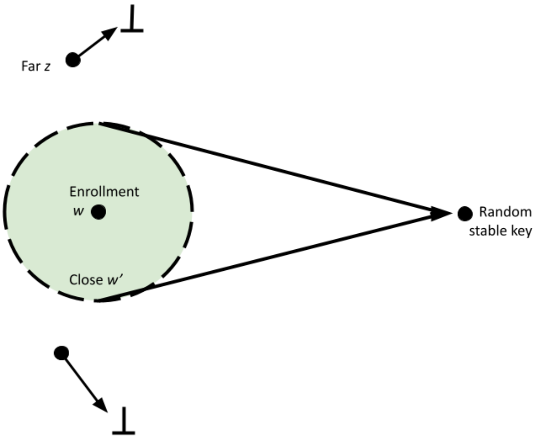


Constructing Nonmalleable Digital Lockers



Robust Fuzzy Extractors in the Plain Model

What is a Fuzzy Extractor?



Robust Fuzzy Extractors

In particular...

- $(key, pub) \leftarrow \mathsf{Gen}(w)$.
- $key' \leftarrow \mathsf{Rep}(pub, w')$

In particular...

- $(key, pub) \leftarrow \mathsf{Gen}(w)$.
- $key' \leftarrow \mathsf{Rep}(pub, w')$

We need...

- $key' = key \iff d(w, w')$ is small.
- No adversary can distinguish *key* given only *pub*.
- No adversary can maul *pub* to reproduce a new key' on some presampled $\widetilde{w}$.

In particular...

- $(key, pub) \leftarrow \mathrm{Gen}(w)$.
- $key' \leftarrow \mathrm{Rep}(pub, w')$

We need...

- $key' = key \iff d(w, w')$ is small.
- No adversary can distinguish *key* given only *pub*.
- No adversary can maul *pub* to reproduce a new key' on some presampled $\widetilde{w}$.

A **Secure Sketch** instead may be thought as recovering w from *pub* and close w':

- $(key, pub) \leftarrow \mathrm{Gen}_{\mathrm{SS}}(w)$.
- $w'' \leftarrow \mathrm{Rep}_{\mathrm{SS}}(pub, w')$

Syndromes and ECCs

Definition

A matrix $\mathrm{Syn} : \mathbb{F}_q^n \to \mathbb{F}_q^{n-k}$ with two properties:

1. $\forall x$ where $|x| \leq t$, $\mathrm{Syn}(x)$ is unique and can be inverted.
2. $\forall s, s'$ where $|s|, |s'|, |s' - s| \leq t$,

$$\begin{aligned}\mathrm{Invert}(\mathrm{Syn}(s' - s)) &= \mathrm{Invert}(\mathrm{Syn}(s') - \mathrm{Syn}(s)) \\ &= \mathrm{Invert}(\mathrm{Syn}(s')) - \mathrm{Invert}(\mathrm{Syn}(s)) \\ &= s' - s\end{aligned}$$

Definition (Syndrome Secure Sketch)

Define $\mathrm{SS}(w) = \mathrm{Syn}(w)$ and

$$\begin{aligned}\mathrm{Rec}(w', s) &= w' - \mathrm{Invert}(\mathrm{Syn}(w') - s) \\ &= w' - \mathrm{Invert}(\mathrm{Syn}(w' - w)) = w.\end{aligned}$$

Then, $(\mathrm{SS}, \mathrm{Rec})$ is a Syndrome Secure Sketch.

Definition (Syndrome Secure Sketch)

Define $\mathsf{SS}(w) = \mathsf{Syn}(w)$ and

$$\begin{aligned}\mathsf{Rec}(w', s) &= w' - \mathsf{Invert}(\mathsf{Syn}(w') - s) \\ &= w' - \mathsf{Invert}(\mathsf{Syn}(w' - w)) = w.\end{aligned}$$

Then, $(\mathsf{SS}, \mathsf{Rec})$ is a Syndrome Secure Sketch.

Definition (Syndrome Secure Sketch)

Define $\mathsf{SS}(w) = \mathsf{Syn}(w)$ and

$$\begin{aligned}\mathsf{Rec}(w', s) &= w' - \mathsf{Invert}(\mathsf{Syn}(w') - s) \\ &= w' - \mathsf{Invert}(\mathsf{Syn}(w' - w)) = w.\end{aligned}$$

Then, $(\mathsf{SS}, \mathsf{Rec})$ is a Syndrome Secure Sketch.

- **Essential idea:** We can find the small shift in w' as a unique syndrome!

Definition (Syndrome Secure Sketch)

Define $\mathsf{SS}(w) = \mathsf{Syn}(w)$ and

$$\begin{aligned}\mathsf{Rec}(w', s) &= w' - \mathsf{Invert}(\mathsf{Syn}(w') - s)\\ &= w' - \mathsf{Invert}(\mathsf{Syn}(w' - w)) = w.\end{aligned}$$

Then, $(\mathsf{SS}, \mathsf{Rec})$ is a Syndrome Secure Sketch.

- **Essential idea:** We can find the small shift in w' as a unique syndrome!
- In particular, can extract the difference in secure sketches by the difference in the Invert of their difference!

Syndrome Secure Sketches

Definition (Syndrome Secure Sketch)

Define $\mathsf{SS}(w) = \mathsf{Syn}(w)$ and

$$\begin{aligned}\mathsf{Rec}(w', s) &= w' - \mathsf{Invert}(\mathsf{Syn}(w') - s) \\ &= w' - \mathsf{Invert}(\mathsf{Syn}(w' - w)) = w.\end{aligned}$$

Then, $(\mathsf{SS}, \mathsf{Rec})$ is a Syndrome Secure Sketch.

- **Essential idea:** We can find the small shift in w' as a unique syndrome!
- In particular, can extract the difference in secure sketches by the difference in the Invert of their difference!
- **Yields robustness!**

Conclusion

Conclusions and Future Directions

Our Results:

- **Defined** a new primitive, nonmalleable point obfuscations with associated data
- **Constructed** the above and the first nonmalleable digital lockers in the plain model
- **Pulled** robust fuzzy extractors with low input entropy into the plain model

Our Results:

- **Defined** a new primitive, nonmalleable point obfuscations with associated data
- **Constructed** the above and the first nonmalleable digital lockers in the plain model
- **Pulled** robust fuzzy extractors with low input entropy into the plain model

Future Directions:

- Plain model nonmalleable obfuscation of other evasive functions such as wildcards, conjunctions, hyperplanes
- Achieving more broad notions of composability/composability of digital lockers
- Constructing reusable plain model fuzzy extractors, other desirable properties
- Other applications of nonmalleable point obfuscation with associated data

Thank you!
Any Questions?