A short report on:

On the Existence of Extractable One-Way Functions

Karim Baghery

Supervised by Michal Zajac

University of Tartu, Estonia

karim.baghery@ut.ee

May 23, 2017

Abstract

Extractability is one of the main and central tools for design and analysis of modern cryptographic protocols. We say a function is extractable if any adversary that outputs a point in the range of the function is guaranteed to "know" a corresponding preimage. This research presents a short report on the existence of extractable one-way functions based on a recent paper by Bitansky et al. [BCPR16]. We discuss with details on two main results of the paper which: 1) shows that if there exist indistinguishability obfuscators for a certain class of circuits, then there do not exist (generalized) extractable one-way functions, with respect to common auxiliary-input of unbounded polynomial length. 2) presents a contraction of bounded-auxiliary-input extractable one-way functions and bounded-auxiliary-input generalized extractable one-way functions based on relatively standard assumptions (e.g., sub-exponential hardness of Learning with Errors). Finally, we conclude the report.

1 Introduction

Extractability plays a central role in cryptographic protocol design and analysis. Basically, it relates to two-party protocols where one of the parties (P) has secret input, and tries to convince the other party (V) that it knows the witnesses or secret. The idea is to argue that if V accepts the argument or proof, then P indeed knows the secret. More precisely, extractability makes the following requirement: Given access to the internals of any (potentially malicious) P, it is possible to explicitly and efficiently compute (extract) the secret value as long as V accepts an interaction. The notion of extractable functions extends the concept of extractability to the more basic setting of computing a function. In this case, convincing a V is replaced by "outputting a value in the range of the function". More specifically, any machine $\mathcal{M}$ that generates a point in the range knows a corresponding preimage in the sense that a preimage is efficiently recoverable given the internal state of the machine [CD09].

Now one may ask, how is the secret or the knowledge extracted? The basic technique for extracting secret or knowledge from an adversary is rewinding which one needs to run
it on multiple related inputs to deduce what it knows from the resulting outputs. This technique treats the adversary as a black-box manner without knowing anything regarding its internals. Fig. 1 shows a graphical representation of extracting by interaction or black-box interaction (rewinding). In several cases impossibility results for black-box reductions and simulation are reported which show that the technique is quite limited. Executing several rounds with the adversary is one of the main limitations of this technique which may not be practical in some cases. As an example, proving security of succinct non-interactive knowledge of arguments (SNARKs) or succinct non-interactive arguments (SNARGs) are out of the techniques reach [GW11, Lip13].

In 1992, Damgård [Dam91] presented an approach to knowledge extraction in the form of knowledge of exponent assumption (KEA). The KEA basically says that it is possible to extract the secret value $x$ from any adversary that, given two random generators $g, h$ of an appropriate group $G$, outputs a pair of $g^x, h^x$. Fig. 2 shows a graphical representation of the KEA assumption. Based on this assumption, the extractable functions have formulated. As already mentioned, extractable functions are function families $\cal F : \{f_e\}$ where, besides standard hardness properties (g.e. one-wayness), any adversary $A$ that given $e$ outputs $y$ in the image of $f_e$ has an extractor $\cal E$ that given $e$ and the code of $A$, outputs a preimage of $y$.

Consequently, extractable functions provide an alternative extraction method which does not need to interaction with $A$. Indeed, it is shown that the KEA or even general extractable one-way functions, are enough for constructing 3-message zero-knowledge protocols, and extractable collision-resistant hash functions are enough for constructing SNARGs [BCCT12].

1.1 Overview on Questions and Results of the Paper

The main contribution of the paper is answering to the following questions,

1. *Can one show that extractable functions cannot exist?*

2. *Can one construct extractable functions from standard hardness assumptions?*
The answers to the questions have presented as two following theorems which will be discussed with more details in the rest of report.

**Theorem A (informal):** If there exist indistinguishability obfuscators for a certain class of circuits, then there do not exist extractable one-way functions with respect to common auxiliary-input of unbounded polynomial length.

**Theorem B (informal):** Assuming one-way functions and publicly-verifiable P-delegation, there exist extractable one-way functions with respect to (common or individual) auxiliary-input of bounded polynomial length.

*Structure of the report:* Section 2 reviews some relevant definitions which are used in the rest of report. In section 3, we discuss on the limitation on unbounded auxiliary-input extractable one-way functions based on indistinguishability obfuscation. Section 4 answers to the second question and explains presented construction of (generalized) bounded-auxiliary-input extractable one-way functions. Finally, we conclude the report in section 5.

## 2 Preliminaries

This section contains the definitions of auxiliary-input extractable one-way functions (EOWFs), bounded-auxiliary-input EOWFs, and generalized extractable one-way functions (GEOWFs) which are three essential concepts for the rest of report.

**Definition 2.1 (Auxiliary-input EOWFs).** Let \( \ell, \ell', m \) be polynomially bounded length functions. An efficiently computable family of functions: \( F = \{ f_e : \{0,1\}^{\ell(n)} \to \{0,1\}^{\ell'(n)} \mid e \in \{0,1\}^m(n), n \in \mathbb{N} \} \), (with an efficient key sampler \( K_F \)), is an auxiliary-input EOWF if it is:

1. **One-way:** For any PPT \( A \), polynomial \( b \), security parameter \( n \in \mathbb{N} \) (large enough), and \( z \in \{0,1\}^b(n) \):
   \[
   \Pr_{e \leftarrow K_F(1^n)} \left[ x' \leftarrow A(e,f_e(x);z) \middle| f_e(x') = f_e(x) \right] \leq \text{negl}(n).
   \]

2. **Extractable:** For any PPT \( A \), there exists a PPT extractor \( E \) such that, for any polynomial \( b \), security parameter \( n \in \mathbb{N} \) (large enough), and \( z \in \{0,1\}^b(n) \):
   \[
   \Pr_{e \leftarrow K_F(1^n)} \left[ y \leftarrow A(e;z) \middle| \exists x : f_e(x) = y \wedge f_e(x') \neq y \right] \leq \text{negl}(n).
   \]

Note that, in the auxiliary-input EOWFs an adversary \( A \) gathers information \( z \) from other components and use it as additional auxiliary input when evaluating the extractable function.

**Definition 2.2 (Bounded auxiliary-input EOWFs).** The definition of the bounded auxiliary-input EOWFs is the same with definition 2.1, but where extraction is guaranteed with respect to auxiliary input of any polysize \( b \), here \( b \) is fixed in advance and the function is
designed accordingly. Which means, extraction is only guaranteed against $b$-bounded adversaries ($z$ is bounded by $b$), however their running time may still be an arbitrary polynomial. Note that, for bounded auxiliary-input EOWFs, a function is key-less if in all the above definitions the key is always set to be the security parameter, namely $e \equiv 1^n$.

**Definition 2.3** (Generalized Extractable One-Way Functions (GEOWFs)). An efficiently computable family of functions, $F = \{ f_e : \{0,1\}^{\ell(n)} \to \{0,1\}^{\ell(n)} \mid e \in \{0,1\}^{m(n)}, n \in \mathbb{N}\}$, (with an efficient key sampler $K_F$), is a GEOWF, with respect to a relation $R^F_e(y, x)$ on triples $(e, y, x) \in \{0,1\}^{m(n)+\ell(x)+\ell(n)}$, if it is:

1. $\mathcal{R}^F$-Hard: For any PPT $A$, polynomial $b$, security parameter $n \in \mathbb{N}$ (large enough), and $z \in \{0,1\}^{b(n)}$:
   \[
   Pr_{e \leftarrow K_F(1^n)} \left[ x' \leftarrow A(e, f_e(x); z) \middle\mid \mathcal{R}^F_e(f_e(x), x') = 1 \right] \leq \text{negl}(n) .
   \]

2. $\mathcal{R}^F$-Extractable: For any PPT $A$, there exists a PPT extractor $E$ such that, for any polynomial $b$, large enough security parameter $n \in \mathbb{N}$, and $z \in \{0,1\}^{b(n)}$:
   \[
   Pr_{e \leftarrow K_F(1^n)} \left[ y \leftarrow A(e; z) \middle\mid \exists x : f_e(x) = y \wedge \mathcal{R}^F_e(f_e(x), x') \neq 1 \right] \leq \text{negl}(n)
   \]

A GEOWF is called:

- **Public-verifiable** if $\mathcal{R}^F_e(f_e(x), x')$ can be efficiently computed by a tester $T(e, f_e(x), x')$.
  In other words, for $(y = f_e(x); x')$, the relation $\mathcal{R}^F_e(y, x')$, can be efficiently tested given $y$ and $x'$ only (and the key $e$ if the function is keyed).

- **Private-verifiable** if $\mathcal{R}^F_e(f_e(x), x')$ can be efficiently computed by a tester $T(e, x, x')$.
  In other words, the relation $\mathcal{R}^F_e(y, x')$, might not be efficiently testable given only $(y = f_e(x), x')$, but it is possible to efficiently test the relation given $x$ in addition.

### 3 Impossibility of Unbounded-Auxiliary-Input EOWFs

This section answers to the first question of the paper which already introduced in section 1. More precisely, the results of this section state that if there exists indistinguishability obfuscation (IO), there do not exist (generalized) auxiliary-input extractable one-way functions. Let us start by defining $iO$ and puncturable PRFs which are two essential preliminaries for the rest of section.

**Definition 3.1** (Indistinguishability Obfuscation). A PPT algorithm $iO$ is said to be an indistinguishability obfuscator (INDO) for $\mathcal{C}$, if it satisfies two main properties including:

1. **Functionality**: For any $C \in \mathcal{C}$: $Pr[\forall x : iO(C)(x) = C(x) = 1]$.  

2. Indistinguishability: For any class of circuit pairs \( \{(C_n^{(1)}, C_n^{(2)}) \in \mathcal{C} \times \mathcal{C} \}_{n \in \mathbb{N}} \), where the two circuits in each pair have the same size and functionality, it holds that:
\[
\{iO(C_n^{(1)})\}_{n \in \mathbb{N}} \approx_c \{iO(C_n^{(2)})\}_{n \in \mathbb{N}}.
\]
In this paper, \( iO \) obfuscators modeled as a game which the challenger samples a pair of circuits \( \mathcal{C}_0, \mathcal{C}_1 \) from a specific distribution \( \mathcal{D} \) and hands them to attacker together with \( iO(C_b) \) and the attacker should guess the bit \( b \).

Definition 3.2 (Puncturable PRFs). Let \( \ell, m \) be polynomially bounded length functions. An efficiently computable family of functions: 
\[
\mathcal{PRF} = \{ \text{PRF}_k : \{0, 1\}^m(n) \rightarrow \{0, 1\}^\ell(n) | k \in \{0, 1\}^n, n \in \mathbb{N} \} \]  
(with a key sampler \( K_{\mathcal{PRF}} \)), is a puncturable PRF if there exists a puncturing algorithm \( \text{Punc} \) that takes as input a key \( k \in \{0, 1\}^n \) and a point \( x^* \), and outputs a punctured key \( k_{x^*} \), so that the following conditions are satisfied:

1. Functionality is preserved under puncturing: For every \( x^* \in \{0, 1\}^{\ell(n)} \),
\[
\Pr_{k \leftarrow K_{\mathcal{PRF}}(1^n)}[\forall x \neq x^* : \text{PRF}_k(x) = \text{PRF}_{k_{x^*}}(x) | k_{x^*} = \text{Punc}(k, x^*)] = 1.
\]

2. Indistinguishability at punctured points: The following ensembles are computationally indistinguishable:
- \( \{x^*, k_{x^*}, \text{PRF}_k(x^*) | k \leftarrow K_{\mathcal{PRF}}(1^n), k_{x^*} = \text{Punc}(k, x^*)\}_{x^* \in \{0, 1\}^m(n), n \in \mathbb{N}} \)
- \( \{x^*, k_{x^*}, u | k \leftarrow K_{\mathcal{PRF}}(1^n), k_{x^*} = \text{Punc}(k, x^*), u \leftarrow \{0, 1\}^{\ell(n)}\}_{x^* \in \{0, 1\}^m(n), n \in \mathbb{N}} \)

It is shown that the GGM PRF [GGM86] yields puncturable PRFs conforming to the above definition.

After reviewing the definitions of the Indistinguishability Obfuscation and Puncturable PRFs which are two basic concepts for this section, we explain first main contribution of the paper with a theorem as follows.

Theorem 1: Assuming indistinguishability obfuscation for all circuits, neither EOWFs nor GEOWFs exist, with respect to common auxiliary-input of unbounded polynomial length.

Note that, there are two points about the theorem which should be highlighted before proving. First, the theorem emphasizes for the case that the adversary and extractor having common auxiliary-input, and the second fact is that the result of the theorem does not apply for any distribution on common auxiliary-inputs, but rather shows that some specific auxiliary-input distribution fails extractability. In particular, they do not rule out natural distributions such as the uniform distribution.

Proof: In order to prove the Theorem, for any EOWF family \( \mathcal{F} \), one needs describe an adversary \( A \) and a distribution \( \mathcal{Z} \) on auxiliary inputs, such that for auxiliary inputs sampled from \( \mathcal{Z} \), any extractor fails.
3.1 The Universal Adversary

We consider a universal PPT adversary $A$ that given $(e; z) \in \{0,1\}^m(n) \times \{0,1\}^{\text{poly}(n)}$, parses $z$ as a circuit and returns $z(e)$.

3.2 The Auxiliary Input Distribution

Let $F$ be a family of extractable one-way functions and let $\mathcal{PRF}$ be a puncturable pseudo-random function family. We start by defining two families of circuits as,

- $\mathcal{C} = \{ C_k : \{0,1\}^m(n) \to \{0,1\}^\ell(n) \mid k \in \{0,1\}^n, n \in \mathbb{N} \}$
- $\mathcal{C}^* = \{ C_{k^*,y^*} : \{0,1\}^m(n) \to \{0,1\}^\ell(n) \mid k \in \{0,1\}^n, e \in \{0,1\}^m(n), y^* \in \{0,1\}^\ell, n \in \mathbb{N} \}$

which are shown in Fig. 3 and Fig. 4. As it can be seen in the Fig. 3 as well, the circuit $C_k$, given a key $e$ for an EOWF, applies $\mathcal{PRF}_k$ to $e$, obtains an input $x$, and returns the result of applying the EOWF $f_e$ to $x$. And similarly, as one can see in the Fig. 4, the circuit $C_{k^*,y^*}$, has a hardwired PRF key $k^*$ that was derived from $k$ by puncturing it at the point $e^*$. In addition, it has hardwired an output $y^*$ to replace the punctured result. In particular, when $y^* = f_{e^*}(\mathcal{PRF}_k(e^*))$ the circuit $C_{k^*,y^*}$ computes the same function as $C_k$ in Fig. 3.

As a next step, let us review the defined auxiliary input distribution $\mathcal{Z} = \{ Z_n \}_{n \in \mathbb{N}}$. Let $s = s(n)$ be the maximal size of circuits in either $\mathcal{C}$ or $\mathcal{C}^*$, corresponding to security parameter $n$, and denote by $[C]_s$ a circuit $C$ padded with zeros to size $s$. Now, by considering $\text{iO}$ as an indistinguishability obfuscator, the distribution $Z_n$ consists of an obfuscated (padded) circuit $C_k$ which can be observed in Fig. 5.
3.3 Adversary $\mathcal{A}$ Does Not Have an Extractor

Now, as the main step of proof, it is shown that $\mathcal{A}$ cannot have any extractor $\mathcal{E}$ satisfying the definition 2.1. Surprisingly, it is shown that for the auxiliary input distribution $Z_n$, any extractor fails with overwhelming probability.

**Lemma 1.** Let $\mathcal{E}$ be any PPT candidate extractor for $\mathcal{A}$ then

$$\Pr_{e \leftarrow K_F(1^n), z \leftarrow Z_n} \left[ y \leftarrow \mathcal{A}(e; z) \quad \exists x : f_e(x) = y \quad x' \leftarrow \mathcal{E}(e; z) \quad f_e(x') \neq y \right] \geq 1 - \text{negl}(n)$$

Note that, since the key $e$ is sampled above independently of the auxiliary input $z$, the above indeed disproves extractability.

**Proof.** To prove, first one needs to show that

$$\Pr_{e \leftarrow K_F(1^n), z \leftarrow Z_n} \left[ y \leftarrow \mathcal{A}(e; z) \quad \exists x : f_e(x) = y \right] = 1 ;$$

which from the definitions of $\mathcal{A}$ and $Z_n$, and the correctness of $iO$, one can get that $\mathcal{A}(e, z) = z(e) = C_k(e) = f_e(\text{PRF}_k(e))$, where $C_k \in \mathcal{C}$ is the circuit obfuscated in $z$, i.e. $z = iO([C_k]_s)$. Now, as a contradiction assume that, for infinitely many $n \in N$, the extractor $\mathcal{E}$ successfully outputs a preimage with noticeable probability $\varepsilon(n)$, i.e.

$$\Pr_{e \leftarrow K_F(1^n), z \leftarrow Z_n} \left[ x' \leftarrow \mathcal{E}(e; z) \quad f_e(x') = z(e) = f_e(\text{PRF}_k(e)) \right] \geq \varepsilon(n) ;$$

where again $z = iO([C_k]_s)$. Next, for every $e^*$, let us consider an alternative distribution $Z_n(e^*, y^*)$ that, instead of sampling a circuit $C_k$, samples a circuit $C_{k_{e^*}, y^*}$, by first sampling $k$ as usual, and then computing $y^* = f_{e^*}(\text{PRF}_k(e^*))$, and the punctured key $k_{e^*}$. (Note that $Z_n(e^*, y^*)$ is actually only parameterized by $e^*$, $y^*$ is added to the notation, to be more explicit.) Now, one can claim that the extractor still succeeds in finding a preimage, i.e.,

$$\Pr_{e^* \leftarrow K_F(1^n), z^* \leftarrow Z_n(e^*, y^*)} \left[ x' \leftarrow \mathcal{E}(e^*; z^*) \quad f_{e^*}(x') = z^*(e^*) = y^* = f_{e^*}(\text{PRF}_k(e^*)) \right] \geq \varepsilon(n) - \text{negl}(n) .$$

which follows from the fact that, for any $e^*$ and $k$, $C_k$ and $C_{k_{e^*}, y^*}$ compute the same function, and the $iO$ indistinguishability guarantee.
In the next step, one needs to consider another experiment where \( Z_n(e^*; y^*) \) is modified to a new distribution \( Z_n(e^*; u) \) that, instead of sampling \( y^* = f_{e^*}(PRF_k(e^*)) \) in \( C_{k_e^*, y^*} \), samples \( y^* = f_{e^*}(r) \), for an independent random \( r \leftarrow \{0, 1\}^\ell \). Then, one can claim that

\[
Pr[\ e^* \leftarrow K_F(1^n), \ z^* \leftarrow Z_n(e^*, y^*) \mid x' \leftarrow E(e^*, z^*) = z^*(e^*) = y^* = f_{e^*}(r) \] \geq \varepsilon(n) - \text{negl}(n).
\]

which, this holds because of the fact that \( PRF_k(e^*) \) is pseudo-random, even given the punctured key \( k_{e^*} \). This means that \( E \) can be used to break the one-wayness of \( F \). Actually, in summary, given a random key \( e^* \), and a challenge \( y^* = f_{e^*}(r) \), an inverter can simply sample a punctured \( k_{e^*} \) on its own, construct the circuit \( C_{k_{e^*}, y^*} \), with its challenge \( y^* \) hardwired in, and sample an obfuscation \( z^* \leftarrow iO(C_{k_{e^*}, y^*}) \). Finally, it runs \( E(e^*, z^*) \) to invert \( y^* \), with the same probability \( \varepsilon(n) - \text{negl}(n) \).

It is worth mentioning that puncturable PRFs can be constructed from one-way functions and EOWF is already a OWF. As a result, the impossibility of auxiliary-input EOWFs is implied only by indistinguishability obfuscation assumption.

### 4 Bounded-Auxiliary-Input EOWFs

This section aims to answer the second question of the paper which already introduced in section 1. As a recall, is it possible to construct extractable functions from standard hardness assumptions?

The constructions of this section, mainly rely on non-interactive universal arguments for deterministic computations. So, similar to last section, before presenting the constructions, let us review the definition of them. Here, \( L_{ul} \) is the universal language consisting of all tuples \((\mathcal{M}, x, t)\) such that \( \mathcal{M} \) accepts \( x \) within \( t \) steps. And \( L_{ul}(T) \) all pairs \((\mathcal{M}, x)\) such that \((\mathcal{M}, x, T) \in L_{ul}\).

Let \( T(n) \) be a computable superpolynomial function. An NIUA system for \( Dtime(T) \) consists of three algorithms \((G, P, V)\) that act as follows. The generator \( G \), given a security parameter \( 1^n \), outputs a reference string \( \sigma \) and a corresponding verification state \( \tau \); in particular, \( G \) is independent of any statement to be proven later. The honest prover \( P(\mathcal{M}, x; \sigma) \) generates a \( \pi \) for the fact that \((\mathcal{M}, x) \in L_{ul}(T(n))\). The verifier \( V(\mathcal{M}, x; \pi, \tau) \) verifies the validity of the \( \pi \). Formally these explanations can be written as follows.

**Definition 4.1 (Non-Interactive Universal Arguments).** A triple \((G, P, V)\) is a non-interactive universal argument system for \( Dtime(T) \) if it satisfies the following properties:

1. **Perfect Completeness:** For any \( n \in \mathbb{N} \) and \((\mathcal{M}, x) \in L_{ul}(T(n))\):

   \[
   Pr[\ V(\mathcal{M}, x; \pi, \tau) = 1 \mid (\sigma, \tau) \leftarrow G(1^n), \ \pi \leftarrow P(\mathcal{M}, x; \sigma) ] = 1
   \]
2. **Adaptive soundness for a bounded number of statements:** There is a polynomial $b$, such that for any poly-size prover $P^*$, large enough $n \in \mathbb{N}$, and set of at most $2^b(n)$ false statements $S \subset \{0, 1\}^{\text{poly}(n)} \setminus \mathcal{L}_{\text{U}}(T(n))$:

$$\Pr[ V(\mathcal{M}, x; \pi, \tau) = 1 \mid (\sigma, \tau) \leftarrow \mathcal{G}(1^n), (\mathcal{M}, x, \pi) \leftarrow P^*(\sigma), (\mathcal{M}, x) \in S ] \leq \text{negl}(n).$$

3. **Fast verification and relative prover efficiency:** There is a polynomial $p$, such that for every $n \in \mathbb{N}$, $t \leq T(n)$ and $(\mathcal{M}, x) \in \mathcal{L}_{\text{U}}(t)$:

- the generator $\mathcal{G}$ runs in time $p(n)$;
- the verifier $V$ runs in time $p(n + |\mathcal{M}| + |x|)$
- the prover $P$ runs in time $p(n + |\mathcal{M}| + |x| + t)$

The system NIUA is said to be publicly-verifiable if soundness is maintained when the malicious prover is also given the verification state $\tau$. In this case, the verification state $\tau$ appears in the clear in the reference string $\sigma$.

**Theorem 2 [KRR14]:** Assuming the Learning with Errors Problem is sub-exponentially hard, for any $b(n) = \text{poly}(n)$, and $T(n) \in (2^{\omega(\log n)}, 2^{\text{poly}(n)})$, there exists a (privately-verifiable) NIUA with adaptive soundness for $2^b(n)$ statements.

### 4.1 The Generalized Extractable One-way Function

This section presents construction of the bounded-auxiliary-input GEOWFs based on any NIUA. Let $b(n)$ be a polynomial and $(\mathcal{G}, P, V)$ be an NIUA system for $\text{Dtime}(T(n))$ for some function $T(n) \in (2^{\omega(\log n)}, 2^{\text{poly}(n)})$, with adaptive soundness for $2^b(n)$ statements. We assume that the system handles statements of the form $(\mathcal{M}, v) \in \{0, 1\}^{b(n)} \times \{0, 1\}^{b(n)+n}$ to prove that “$\mathcal{M}(1^n)$ outputs $v$ in $T(n)$ steps”. Assume that, $\mathcal{G}(1^n; r)$ uses randomness of size $n$ to output a reference string of polynomial size $m(n)$, and a verification state $\tau$ (if the system is publicly-verifiable, then $\tau$ appears in $\sigma$). Consider that $P$ outputs certificates $\pi$ of size $p(n)$. Let PRG be a pseudo random generator stretching $n$ bits to $b(n) + n$ bits. They construct a key-less family of functions $\mathcal{F} = \{f_n\}_{n \in \mathbb{N}}$, consisting of one function $f_n : \{0, 1\}^{\ell(n)} \rightarrow \{0, 1\}^{\ell'(n)}$, for each security parameter $n$, where $\ell(n) = \max(2n, b(n) + p(n))$ and $\ell'(n) = m(n) + b(n) + n$. The structure of the function is shown in Fig. 6.
Then they define the corresponding relation \( \mathcal{R}_F = \{ R_n \} \) in Fig. 7, which will be publicly verifiable (respectively, privately-verifiable) if the NIUA is publicly (respectively, privately verifiable). We assume that the NIUA is such that for every valid reference string \( \sigma \) produced by \( G \), there is a single possible verification state \( \tau \) (this can always be achieved by adding a commitment to \( \tau \) inside \( \sigma \)).

**Theorem 3:** The function family \( F = \{ f_n \} \) given in the Fig. 6 is a GEOWF, with respect to \( \mathcal{R}_F \) against \((b(\log n) - \omega(1))\)-bounded auxiliary-input.

**Sketch proof:** Firstly, in order to break \( \mathcal{R}_F \)-hardness, an adversary given a random image \((\sigma, \bar{v})\) (where \( v = PRG(s) \) is of length \( b(n) + n \)) has to come up with a small machine \( M \), whose description length is at most \( b(n) \), and a proof that \( M \) outputs \( v \) (within a \( T(n) \) steps). However, in an indistinguishable world where \( v \) is a truly random string, \( v \) would almost surely have high Kolomogorov complexity, and a short machine \( M \) that outputs \( v \) would not exist. Thus, in this case, the breaker has to produce an accepting proof for a false statement, and violate the soundness of the NIUA, so \( F \) is \( \mathcal{R}_F \)-hard.

As for extraction, given a poly-time machine \( M_z \) with short advice \( z \) that outputs \((\sigma, v)\), where \( \sigma \) is a valid reference string for the NIUA system, the extractor simply computes a proof \( \pi \) for the fact that \( M_z \) outputs \( v \), and outputs the witness \((M_z, \pi; pad)\). By the completeness of the NIUA system, the proof \( \pi \) is indeed accepting, and the witness satisfies \( \mathcal{R}_F \). Furthermore, by the relative prover efficiency of the NIUA, the extractor runs in time that is polynomial in the running time of the adversary \( M_z \).

**Proof.** To prove formally, one needs to show \( \mathcal{R}_F \)-hardness and \( \mathcal{R}_F \)-extractability.

**\( \mathcal{R}_F \)-hardness:** Suppose that there exists a breaker \( B \) that, given \( y = (\sigma, v) \), where \( \sigma \leftarrow G(1^n) \), and \( v \leftarrow PRG(U_n) \), finds \( x = (M, \pi, pad) \) such that \( \mathcal{R}_F(y, x) = 1 \) with noticeable probability \( \varepsilon(n) \). So, one can construct a malicious prover \( P^* \) that breaks the adaptive soundness of the NIUA (for \( 2^{b(n)} \) statements), with probability \( \varepsilon(n) - \text{negl}(n) \). \( P^* \), given \( \sigma \), first samples on its own \( \hat{v} \leftarrow U_{b(n)+n} \) (independently of \( \sigma \)), and then runs \( B(\sigma, \hat{v}) \) to obtain a machine \( M \) of size \( b(n) \), and a proof \( \pi \).

In this state, one may claim that with probability \( \varepsilon(n) - \text{negl}(n) \), \( \pi \) is an accepting proof for the statement \((M, \hat{v})\) asserting that ”\( M(1^n) \) outputs \( \hat{v} \) in \( T(n) \) steps”. Indeed, the view of \( B \) in the above experiment is identical to its real view, except that it gets a truly random
\( \hat{v} \), rather than a pseudo-random \( v \) that was generated using PRG. Thus, the claim follows by the PRG guarantee.

Next, one can claim that since \( \hat{v} \) is a \((b(n) + n)\)-long random string, except with negligible probability \( 2^{-n} \), there does not exist \( \mathcal{M} \) of size \( b(n) \) that outputs \( \hat{v} \). Thus, \( \mathcal{P}^* \) produces an accepting proof for one of \( 2^{b(n)} \) false statements given by the adaptive choice of \( \mathcal{M} \in \{0, 1\}^{b(n)} \), and violates the soundness of the NIUA.

**\( \mathcal{R}^F \)-extractability:** Next step is showing \( \mathcal{R}^F \)-extractability. To this aim, one can that there is one universal PPT extractor \( \mathcal{E} \) that can handle and PPT adversary \( \mathcal{M} \) with advice of size at most \( b(n) - \omega(1) \). For an adversarial code \( \mathcal{M} \) and advice \( z \in \{0, 1\}^{gb(n) - \omega(1)} \), denote by \( \mathcal{M}_z \) the machine that, on input \( 1^n \), runs \( \mathcal{M}(1^n; z) \). The extractor \( \mathcal{E} \) is given \((\mathcal{M}; z)\), where \( \mathcal{M}_z \) has description size at most \( b(n) \) and running time at most \( t_M < T(n) \), and \( \mathcal{M}_z(1^n) = y = (\sigma, v) \in \text{Image}(\mathcal{f}n) \). To compute a witness \( x' \in \mathcal{R}^F(y) \), \( \mathcal{E} \) computes a \( \pi \) for the fact that "\( \mathcal{M}_z(1^n) = v' \)" and then outputs \( x' = (\mathcal{M}_z, \pi, \text{pad}) \). The fact that \( x' \) is indeed a valid witness follows directly from the perfect completeness of the scheme. Finally, it can be seen that by the relative prover efficiency of the NIUA the extractor runs in time that is polynomial in the running time \( t_M \) of the adversary.

\[ \square \]

### 4.2 The Standard Extractable One-way Function

Similar to the last subsection, one can construct a standard extractable one-way function based on publicly-verifiable NIUAs that have an additional property which says: in addition to perfect completeness for an honestly chosen reference string \( \sigma \) (which in the publicly-verifiable case is also the verification state), it is also possible to check whether any given \( \sigma \) is valid, or more generally admits perfect completeness. It is worth mentioning that that exiting candidates for publicly-verifiable NIUAs indeed have this property.

**Definition** (NIUA with key validation). A publicly-verifiable NIUA system is said to have key validation if there exists an efficient algorithm \( \text{Valid} \), such that for any \( \sigma \in \{0, 1\}^{m(n)} \), if \( \text{Valid}(\sigma) = 1 \), then the system has perfect completeness with respect to \( \sigma \). That is, proofs for true statements, generated and verified using \( \sigma \), are always accepted.

Construction of the standard EOWF is similar to previous one, and at a very high-level attempts to embed the structure of the previous GEOWF function and relation into a standard EOWF. Suppose \( b(n) \) be a polynomial and \((\mathcal{G}, \mathcal{P}, \mathcal{V})\) be an NIUA system with the same parameters as in the above GEOWF construction, and with the additional key-validation property. Suppose PRG be a pseudo random generator stretching \( n \) bits to \( b(n) + n \) bits.

They constructed a key-less family of functions \( \mathcal{F} = \{f_n\}_{n \in \mathbb{N}} \), consisting of one function \( f_n : \{0, 1\}^{\ell(n)} \to \{0, 1\}^{\ell'(n)} \), for each security parameter \( n \), where \( \ell(n) = 4n + 2b(n) + m(n) + p(n) \) and \( \ell'(n) = m(n) + b(n) + n \). The function is given in Fig. 8.

**Theorem 4:** The function family \( \mathcal{F} = \{f_n\}_{n \in \mathbb{N}} \), given in the Fig. 8 is an EOWF, against \((b(n) - \omega(1))\)-bounded auxiliary-input.
Figure 8: The function $f_n$

**Sketch proof:** In order to see that $F$ is one-way, except with negligible probability, a random image comes from the "normal branch of the function", where $i \neq \{0^n, 1^n\}$ and includes an honestly sampled $\sigma$ and a pseudorandom string $v = PRG(s)$. And to invert it, an adversary must either invert $PRG(s)$, allowing it to produce a "normal branch" preimage, or obtain a short machine $M$ and an accepting proof $\pi$, that $M$ outputs $v$, allowing it to produce a "trapdoor branch" preimage. In the initial case, the inverter violates the one-wayness of $PRG$. In the second case, similar to the last theorem, the inverter can be used to break the soundness of the NIUA (leveraging the fact that a truly random $\hat{v}$ almost surely cannot be computed by a short machine).

As for extraction, given a poly-time machine $M_z$ with short advice $z$ that outputs $(\sigma, v) \neq \bot$, by the definition of $f_n$, $\sigma$ is a valid reference string for the NIUA system (indeed, $\bot$ is an image that indicates an improper reference string $\sigma$, or a non-accepting proof $\pi$). In this case, the extractor simply computes a proof $\pi$ for the fact that $M_z$ outputs $v$, and outputs the preimage $(0^n; (0^n; 0^n), (\sigma, M_z, v, \pi))$. By the completeness of the NIUA system, for a valid $\sigma$, the proof $\pi$ is indeed accepting. By the relative prover efficiency of the NIUA, the extractor runs in time that is polynomial in the running time of the adversary $M_z$. The only other case to consider is where $M_z$ outputs $\bot$, in which case producing a preimage is easily done by setting $i = 1^n$.

**Proof.** In order to formally proof the theorem, similar to the last theorem, one needs to show $R^F$-hardness, and $R^F$-extractability.  

**One-wayness:** Assume there exists an inverter $I$ that, given $y = f_n(x)$, where $x \leftarrow U_{\ell(n)},$
finds a preimage \( x' = (i', (s', r')) \) with noticeable probability \( \varepsilon(n) \). We construct a prover \( \mathcal{P}^* \) that breaks the adaptive soundness of the NIUA (for \( 2^{b(n)} \) statements), with probability \( \varepsilon(n) - \text{negl}(n) \). \( \mathcal{P}^* \) is defined as in the proof of Theorem 3: given \( \sigma \), it first samples on its own \( \hat{\nu} \leftarrow U_{b(n)+n} \) (independently of \( \sigma \)), and then runs \( \mathcal{I}(\sigma; \hat{\nu}) \) to obtain \( x' = (i', (s', r')), (\sigma', \mathcal{M}', v', \pi')) \).

**Lemma 2.** With probability \( \varepsilon(n) - \text{negl}(n) \), \( \pi' \) is an accepting proof, with respect to \( \sigma \), for the statement \( (\mathcal{M}', v) \), proves that \( \mathcal{M}'(1^n) \) outputs \( \hat{\nu} \) in \( T(n) \) steps”.

The proof of Lemma will conclude the proof of one-wayness since, as in the proof of Theorem 3, except with negligible probability, there does not exist a machine \( \mathcal{M}_0 \) of size \( b(n) \) that outputs \( \hat{\nu} \) which is a \( (b(n) + n) \)-long random string. This means that \( \mathcal{I} \) outputs an accepting proof for one of \( 2^{b(n)} \) false statements (given different \( \mathcal{M} \in \{0, 1\}^{b(n)} \)), and violates the soundness of the NIUA.

**Proof of Lemma 2:** To prove the lemma, one first can consider an hybrid experiment where \( \mathcal{I} \) samples a pseudo random \( v \leftarrow \text{PRG}(U_n) \) instead of a truly random \( \hat{\nu} \). By the PRG guarantee, one can show that the probability of outputting \( (\mathcal{M}_0, \pi) \) as required by the lemma changes at most by a negligible amount \( \text{negl}(n) \). Next, one can get that the view of \( \mathcal{I} \) in the hybrid experiment is identical to its view in the real world where it receives a random image \( y = (\sigma; v) \). Furthermore, whenever \( \mathcal{I} \) finds a preimage \( x' = (i', (s', r'), (\sigma'; \mathcal{M}', v', \pi')) \) of \( y \) such that \( i' = 0^n \), by the definition of \( f_n \), \( (\sigma'; v') = (\sigma; v) \), and \( \pi' \) must be an accepting proof for the statement \( (\mathcal{M}', v' = v) \), with respect to \( \sigma' = \sigma \).

Now, since \( \mathcal{I} \) inverts the function with probability \( \varepsilon(n) \), it thus suffices to show that the preimage it finds is such that \( i = 0^n \), except with negligible probability. Indeed, whenever \( \mathcal{I} \) finds a preimage such that \( i' \not\in \{0^n, 1^n\} \), by the definition of \( f_n \), it inverts \( v = \text{PRG}(s) \), contradicting the one-wayness of PRG. Also, a preimage of \( (\sigma; v) \) cannot have \( i' = 1^n \), assuming \( (\sigma; v) \not\perp \), which is the case with overwhelming probability. This concludes the proof of the lemma.

**\( \mathcal{R}^2 \)-extractability:** For this case, one can show that there is one universal PPT extractor \( \mathcal{E} \) that can handle and PPT adversary \( \mathcal{M} \) with advice of size at most \( b(n) - \omega(1) \). Similar to the proof of theorem 3, for an adversarial code \( \mathcal{M} \) and advice \( z \in \{0, 1\}^{b(n) - \omega(1)} \), denoted by \( \mathcal{M}_z \) the machine that, on input \( 1^n \), runs \( \mathcal{M}(1^n; z) \). The extractor \( \mathcal{E} \) is given \( (\mathcal{M}; z) \), where \( \mathcal{M}_z \) has description size at most \( b(n) \) and running time at most \( t_\mathcal{M} < T(n) \), and \( \mathcal{M}_z(1^n) = (\sigma, v) \in \text{Image}(f_n) \).

If \( (\sigma; v) \neq (0^{m(n)}; 0^{b(n)+n}) \), we know that \( \sigma \) must be valid, in which case \( \mathcal{E} \) computes a \( \pi \) for the fact that \( "\mathcal{M}_z(1^n) = v" \), and then outputs the preimage \( x' = (0^n, (0^n, 0^n), (\sigma, \mathcal{M}_z, v, \pi)) \). The fact that \( x' \) is indeed a valid preimage follows directly from the perfect completeness of the scheme, for a valid \( \sigma \). If \( (\sigma; v) = (0^{m(n)}; 0^{b(n)+n}) \), the extractor outputs the preimage \( x' = (1^n, (0^n, 0^n), (0^{m(n)}, 0^{b(n)}, 0^{b(n)+n}, 0^{b(n)})) \). Finally, it is worth mentioning that by the relative prover efficiency of the NIUA the extractor runs in time that is polynomial in the running time \( t_\mathcal{M} \) of the adversary. \( \square \)
5 Conclusion

Due to essential role of extractability in analysis and security proofs of cryptographic protocols, it is one of the most challenging topics for cryptographers. One of the main applications of extractability is on extractable one-way functions which are probabilistic functions that reveal no information about their input, other than the ability to verify guesses. More precisely, an extractable one-way function, $f$, guarantees that any party that manages to compute a value in the range of $f$ knows a corresponding preimage. The notion of "knowing of preimage" is captured by way of algorithmic extraction.

There are mainly two main variants of extractability, namely non-interactive (non-black box) and interactive (black box or rewinding). The noninteractive variant (i.e., the variant that requires non-interactive extraction) can be regarded as a generalization from specific knowledge assumptions to a notion that is formulated in general computational terms. The interactive- extraction variant treats with adversary in a black-box manner and extracts knowledge from the adversary by sending queries and rewinding.

In this report, we discussed with details about two recent interesting results of Bitansky et al. [BCPR16] regard to existence of the extractable one-way functions. We observe that if there exist indistinguishability obfuscators, then there do not exist (generalized) extractable one-way functions, with respect to common auxiliary-input of unbounded polynomial length. We saw that the main intuition of this result was from [HT98]s which stated that if we can have an arbitrary auxiliary information then this info may include some very obfuscated code or circuit which consequently in order to extract preimage from this circuit, the extractor has to efficiently reverse-engineer which is not possible in some cases. We observed that how by assuming indistinguishability obfuscators how for a universal adversary and for some special distributions of unbounded auxiliary information any extractor fails to extracts.

Then, as a second main result of the paper, we explained the presented construction of (generalized) bounded-auxiliary-input extractable one-way functions which were based on relatively standard assumptions (e.g., sub-exponential hardness of Learning with Errors). To summarize the basic idea behind the presented construction of generalized extractable one-way functions, consider the case that the generalized extractable one-way function $f$ is a key-less and simply a pseudorandom generator stretching inputs of length $n$ to outputs of length $2n$. The relation $R^f$ contains pairs $(y, M)$ such that the witness $M$ is a description of a machine of length at most $n$, and $M(1^n)$ outputs $y$. The fact that the relation $R^f(y, \cdot)$ is hard to satisfy for $y = f(x)$ and a random $x$, follows from the pseudo-randomness of the output $y$. Indeed, a truly random output that is indistinguishable from $y$ would have high Kolmogorov complexity. However, given any adversarial program $M_A$ whose description size is bounded by $n$ and that outputs some $y \in \{0,1\}^{2n}$, the description of the program $M_A$ itself is a witness that satisfies the relation $R^f(M_A)$, and thus extraction is trivial.
References


