Puncturable Witness Pseudorandom Functions and its Applications on Witness Encryption

Tapas Pal, Ratna Dutta

Department of Mathematics,
Indian Institute of Technology Kharagpur,
Kharagpur-721302, India
tapas.pal@iitkgp.ac.in,ratna@maths.iitkgp.ernet.in

Abstract. In this work, we propose a slightly stronger variant of witness pseudorandom function (WPRF) defined by Zhandry (TCC 2016), that we call *puncturable witness pseudorandom function* (pWPRF). It is capable of generating a pseudorandom value corresponding to every statement of an NP language. We utilize the punctured technique to extend applications of WPRF. Specifically, we construct a *semi-adaptively* secure *offline witness encryption* (OWE) scheme using a pWPRF, an indistinguishability obfuscation ($i\mathcal{O}$) and a symmetric-key encryption (SKE), which enables us to encrypt messages along with NP statements. We show that replacing $i\mathcal{O}$ with extractability obfuscation, the OWE turns out to be an *extractable offline witness encryption* scheme. To gain finer control over data, we further demonstrate how to convert our OWEs into *offline functional witness encryption* (OFWE) and *extractable* OFWE. The ciphertext size of current available OWEs grows polynomially with the size of messages, whereas all of our OWEs produce optimal size ciphertexts. Finally, we show that the WPRF of Pal et al. (ACISP 2019) can be extended to a pWPRF and an *extractable* pWPRF.

Keywords: witness pseudorandom function, witness encryption, functional witness encryption, obfuscation.

1 Introduction

Witness Pseudorandom Function. In a usual pseudorandom function, we generate a pseudorandom value for an input $x \in \mathcal{X}$ using a secret-key. Zhandry [16] proposed an enhanced primitive called *witness pseudorandom function* (WPRF) which produces pseudorandom values corresponding to statements of an NP language L with a relation $R : \mathcal{X} \times \mathcal{W} \to \{0, 1\}$. If $x \in L$ then there exists a witness $w \in \mathcal{W}$ such that $R(x, w) = 1$, otherwise R maps to 0. In setup of WPRF, we generate two keys: a secret function key fk and a public evaluation key ek. To compute a pseudorandom value $y \in \mathcal{Y}$ corresponding to a statement $x \in \mathcal{X}$, we use the secret function key fk. The same pseudorandom value y can only be recovered using the public evaluation key ek if $x \in L$ and a witness w is known such that $R(x, w) = 1$. For security we require that y is completely uniform over $\mathcal{Y}$ if $x \notin L$. In *extractable* WPRF, we relax the requirement on security and allow x to be in L. However, in such a scenario, if an adversary can distinguish the honestly computed y from a uniformly chosen element of $\mathcal{Y}$ then we can extract a valid witness of x using an efficient extractor.

Applications of WPRF. We have seen a list of cryptographic primitives realized from WPRF in [16] such as multiparty non-interactive key exchange without trusted setup, poly-many hardcore bits for one-way functions and secret sharing for monotone NP languages. More interestingly, WPRF can be considered a generalization of a modern primitive called witness encryption (WE) [12] which encrypts messages with respect to a NP statement and recovery of the original message from a ciphertext needs a witness for the statement. It has been observed that WPRF directly implies WE. Furthermore, one can construct more refined variant of WE, termed as reusable WE [16], using WPRF. The main goal of reusable WE was to make the encryption algorithm relatively efficient and ciphertext size *optimal*, besides it provides security in chosen ciphertext attack model. On the other hand, extractable WPRF can be used to build a fully distributed broadcast encryption [16] where the size of secret-keys, public-keys and ciphertexts are all poly-logarithmic in the number of users.

Our Contribution. Having seen a series of applications of WPRF in [16], we are keen to explore it more. It is desirable to build a relatively closer primitive such as offline witness encryption (OWE) [1] using WPRF maintaining the same encryption efficiency of the reusable WE. An OWE is more preferable over the normal WE because the computationally hard work is shifted from the encryption algorithm by introducing an additional setup phase. Unfortunately, WPRF does not immediately achieve OWE or offline functional WE [6]. Further, existing OWEs [1,14,8] do not have optimal ciphertext size as in reusable WE of [16].

In this work, we extend the applications of WPRF by introducing a puncturing technique akin to puncturable pseudorandom function (pPRF) [15]. In the security model of normal WPRF, an adversary $\mathcal{A}$ is given access to an oracle $\mathsf{F}(\mathsf{fk}, \cdot)$ which on input $x \in \mathcal{X}$ of $\mathcal{A}$'s choice outputs a pseudorandom value corresponding to x. Naturally, $\mathcal{A}$ is restricted to query on the challenge statement x^* which is not in L. In our setting, instead of giving an access to $\mathsf{F}(\mathsf{fk}, \cdot)$, $\mathcal{A}$ is provided with a punctured key fk_{x^*} which enables $\mathcal{A}$ to learn the pseudorandom value corresponding to any x except x^*. The WPRF is secure if $\mathcal{A}$ is unable to distinguish $\mathsf{F}(\mathsf{fk}, x^*)$ from a random element. We call this variant of WPRF a *puncturable* WPRF (pWPRF). In *extractable* pWPRF, we allow x^* to be in L. In that case, there exists an extractor $\mathcal{E}$ which outputs a witness of x^* with high probability and the run time of $\mathcal{E}$ depends on the distinguishing advantage of $\mathcal{A}$ between $\mathsf{F}(\mathsf{fk}, x^*)$ and a random element. A pWPRF having this extractability property is called *puncturable witness-extractable pseudorandom function* (pWEPRF).

Both WE and WPRF have been realized using various assumptions on multilinear maps [12,16], but recent attacks on multilinear maps [7,9] introduce threats on the security of those schemes. We bring the punctured program technique of PRF [15] in case of WPRF. The main idea is to build two equivalent programs P and P′ where P uses the secret-key oblivious to the adversary and

P' uses a punctured key available to the adversary. An important tool in this setup is indistinguishability obfuscation ($i\mathcal{O}$) [11]. Although, $i\mathcal{O}$ is yet to realize in practical devices, the recent developments in this area [2,3] raise confidence in exploring more usability of $i\mathcal{O}$. We build following primitives using the additional punctured technique of WPRF:

– We build a semi-adaptively secure OWE scheme (Sec. 3) using a pWPRF, an $i\mathcal{O}$, a pseudorandom generator (PRG) and a symmetric key encryption (SKE) scheme. Our OWE achieves optimal ciphertext-size, namely $|m|+\text{poly}(\lambda)$ where $|m|$ is the size of message and λ is the security parameter.
– Replacing $i\mathcal{O}$ with extractability obfuscation ($e\mathcal{O}$) [6], we convert the OWE into an *extractable* OWE (EOWE) in Sec. 3. The ciphertext-size remains the same which is optimal for any public-key encryption scheme.
– An user having a valid witness is able to learn the whole message in normal OWE. This all-or-nothing type encryption may not be sufficient for applications where we need fine-grained access control over the data. In such a scenario, *offline functional* WE (OFWE), introduced by Boyle et al. [6], can be utilized as the user having a valid witness can now learn a function of the message and witness. In this work, we show that our techniques of achieving OWE can be extended to realize semi-adaptively secure OFWE and selectively secure *extractable* OFWE schemes (Sec. 4).
– Next, we show that the WPRF of [14] satisfies our definition of pWPRF (Sec. 5). In particular, we can construct pWPRF using a pPRF and an $i\mathcal{O}$. Furthermore, a pWEPRF can be achieved by replacing the $i\mathcal{O}$ with an $e\mathcal{O}$.

Applications of OWE. There are several applications of WE discussed in [12] such as identity-based encryption (IBE), attribute based encryption (ABE) for circuits. These applications become more efficient with OWE as IBE and ABE both support a trusted setup and hence we can generate the parameters of OWE in their setup phases, which makes the encryption more efficient. Additionally, we can achieve optimal size ciphertexts for these primitives with our OWE. In an asymmetric password based encryption [5], OWE plays an important role to make the encryption more efficient and using our OWE yields optimal size ciphertext. Moreover, a semi-adaptively secure OWE is much more appreciated in time-lock encryption [13] and encryption with puzzle where the statement is fixed once and for all.

Related Works. Zhandry [16] constructed WPRF from subset-sum Diffie-Hellman assumption related to multilinear maps. Getting a pseudorandom value using an evaluation key is computationally expensive as one need to apply a multilinear map with linearity much larger than the size of the NP relation. On the other hand, we extend the $i\mathcal{O}$-based WPRF of [14] into a puncturable WPRF to enhance the field of application. We note that, although obfuscation itself is a powerful assumption, a wide range of functionalities, including the function classes required in this work, can be efficiently realized using Trusted Execution Environments (TEEs), Intel's Software Guard Extensions (SGXs) [4,10].

Abusalah et al. [1] introduced OWE with a purpose of making encryption much more efficient than the existing WEs. However, the OWE of [1] is selectively secure and the size of ciphertexts are not promising as it contains a simulation sound non-interactive zero knowledge proof along with two (public-key) encryptions of the same message. Recently, OWE with semi-adaptive security is built in [8], but the size of ciphertext is not as compact as one would have wanted for light weight devices. On the contrary, our OWEs deliver semi-adaptive security with an optimal size ciphertext similar to the reusable WE of [16].

2 Preliminaries

2.1 Notations

We denote $\lambda \in \mathbb{N}$ by a security parameter. If $x \in \{0,1\}^*$, then we denote $|x|$ by size of the string x. For any set S, the notation $x \leftarrow S$ denotes the process of sampling x uniformly at random from the set S. Let Algo be a probabilistic polynomial time (PPT) algorithm, then $y \leftarrow \mathsf{Algo}(x)$ denotes the execution of Algo with an input x using a fresh randomness and assign the output to y. If the randomness, say r, is provided externally then we denote this execution by $y \leftarrow \mathsf{Algo}(x; r)$. We call $\{\mathcal{C}_\lambda\}$ as a family of polynomial sized circuits if there exists a fixed polynomial p such that $|C| < \mathsf{p}(\lambda)$ for any $C \in \mathcal{C}_\lambda$. We say $\mathsf{negl}\colon \mathbb{N} \rightarrow \mathbb{R}$ be a negligible function of λ if for every positive polynomial p, there exists an integer $n_\mathsf{p} \in \mathbb{N}$ such that $\mathsf{negl}(\lambda) < 1/\mathsf{p}(\lambda)$ for all $n > n_\mathsf{p}$.

2.2 Pseudorandom Generator

Definition 1 A pseudorandom generator (PRG) is a deterministic polynomial time algorithm PRG that on input a seed $s \in \{0,1\}^\lambda$ outputs a string of length $\ell(\lambda)$ such that the following holds:

- *expansion*: For every λ it holds that $\ell(\lambda) > \lambda$.
- *pseudorandomness*: For all PPT adversary $\mathcal{A}$ and $s \leftarrow \{0,1\}^\lambda, r \leftarrow \{0,1\}^{\ell(\lambda)}$ there exists a negligible function negl such that

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{PRG}}(\lambda) = |\Pr[\mathcal{A}(1^\lambda, \mathsf{PRG}(s)) = 1] - \Pr[\mathcal{A}(1^\lambda, r) = 1]\,| < \mathsf{negl}(\lambda).$$

2.3 Puncturable Pseudorandom Function

Definition 2 A puncturable pseudorandom function (pPRF) is a tuple of PPT algorithms (Gen, $\mathsf{PuncKey}$, Eval, $\mathsf{PuncEval}$) defined as follows:

- $\mathsf{K} \leftarrow \mathsf{Gen}(1^\lambda)$: It is a randomized algorithm that takes as input a security parameter λ, and outputs a secret-key K.
- $\mathsf{K}_x \leftarrow \mathsf{PuncKey}(\mathsf{K}, x)$: It is a deterministic algorithm that takes as input a secret-key K and an element $x \in \mathcal{X}$, and produces a punctured key K_x.
- $y \leftarrow \mathsf{Eval}(\mathsf{K}, x)$: It is a deterministic algorithm that takes input a secret key K and an element $x \in \mathcal{X}$, and produces a pseudorandom value $y \in \mathcal{Y}$.
- $\mathsf{PuncEval}(\mathsf{K}_x, x') \in \mathcal{Y} \cup \{\perp\}$: It is a deterministic algorithm that takes as input a punctured key K_x corresponding to some element $x \in \mathcal{X}$ and an element $x' \in \mathcal{X}$, and outputs a pseudorandom value $y \in \mathcal{Y}$ if $x \neq x'$. It outputs $\perp$ if $x = x'$.

The pPRF is said to be correct if the following holds:
– *correctness*: For all distinct pair of elements $x, x' \in \mathcal{X}^2$, $\mathsf{K} \leftarrow \mathsf{Gen}(1^\lambda)$, we require that

$$\Pr[\mathsf{Eval}(\mathsf{K}, x') = \mathsf{PuncEval}(\mathsf{PuncKey}(\mathsf{K}, x), x')] = 1$$

Definition 3 A puncturable pseudorandom function (pPRF) is said to be secure (or preserves pseudorandomness at punctured point) if, for all PPT adversary $\mathcal{A}$ and any $x \in \mathcal{X}$, $\mathsf{K} \leftarrow \mathsf{Gen}(1^\lambda)$, $\mathsf{K}_x \leftarrow \mathsf{PuncKey}(\mathsf{K}, x)$ there exists a negligible function negl such that

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{pPRF}}(\lambda) = |\Pr[\mathcal{A}(1^\lambda, \mathsf{K}_x, \mathsf{Eval}(\mathsf{K}, x)) = 1] - \\ \Pr[\mathcal{A}(1^\lambda, \mathsf{K}_x, y \leftarrow \mathcal{Y}) = 1]\ | < \mathsf{negl}(\lambda).\end{aligned}$$

2.4 Symmetric Key Encryption

Definition 4 A symmetric key encryption (SKE) scheme is a tuple of PPT algorithms (Gen, Enc, Dec) defined as follows:
- $\mathsf{K} \leftarrow \mathsf{Gen}(1^\lambda)$: It is a randomized algorithm that takes as input a security parameter λ and outputs a key K.
- $c \leftarrow \mathsf{Enc}(\mathsf{K}, m)$: It is a deterministic algorithm that takes input a key K and a message $m \in \mathcal{M}$, and produces a ciphertext c.
- $\mathsf{Dec}(\mathsf{K}, c) \in \mathcal{M} \cup \{\perp\}$: It is a deterministic algorithm that takes as input a key K and a ciphertext c, and outputs either a message $m \in \mathcal{M}$ or $\perp$.

The SKE is said to be correct if the following holds:
– *correctness*: For all $m \in \mathcal{M}$ and $\mathsf{K} \leftarrow \mathsf{Gen}(1^\lambda)$, we require that

$$\Pr[\mathsf{Dec}(\mathsf{K}, \mathsf{Enc}(\mathsf{K}, m)) = m] = 1$$

Definition 5 A symmetric key encryption SKE is said to satisfy ciphertext indistinguishability (CIND) security if, for all PPT adversary $\mathcal{A}$ and any pair of equal length messages (m_0, m_1) there exists a negligible function negl such that

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{SKE}}(\lambda) = |\Pr[\mathcal{A}(1^\lambda, \mathsf{Enc}(\mathsf{K}, m_0)) = 1] - \\ \Pr[\mathcal{A}(1^\lambda, \mathsf{Enc}(\mathsf{K}, m_1)) = 1]\ | < \mathsf{negl}(\lambda)\end{aligned}$$

2.5 Puncturable Witness Pseudorandom Function

Definition 6 A puncturable witness pseudorandom function (pWPRF) for an NP language L with a relation R is a tuple of PPT algorithms (Gen, F, $\mathsf{PuncKey}$, PuncF, Eval) defined as follows:
- $(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{Gen}(1^\lambda, R)$: It is a randomized algorithm that takes as input a security parameter λ and a relation circuit $R : \mathcal{X} \times \mathcal{W} \rightarrow \{0, 1\}$, and produces a secret function key fk and a public evaluation key ek.
- $y \leftarrow \mathsf{F}(\mathsf{fk}, x)$: It is a deterministic algorithm that takes input a function key fk and an element $x \in \mathcal{X}$, and produces a pseudorandom value $y \in \mathcal{Y}$.

1. $x^* \leftarrow \mathcal{A}(1^\lambda)$
2. $(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{Gen}(1^\lambda, R)$
3. $\mathsf{fk}_{x^*} \leftarrow \mathsf{PuncKey}(\mathsf{fk}, x^*)$
4. $y_0 \leftarrow \mathsf{F}(\mathsf{fk}, x^*),\ y_1 \leftarrow \mathcal{Y}$
5. $b \leftarrow \{0,1\}$
6. $b' \leftarrow \mathcal{A}(\mathsf{ek}, \mathsf{fk}_{x^*}, y_b)$
7. return 1 if $(b' = b) \wedge (x^* \notin L)$

Fig. 1: $\mathsf{Expt}_{\mathcal{A}}^{\mathsf{pWPRF},R}(1^\lambda)$

1. $x^* \leftarrow \mathcal{A}(1^\lambda)$
2. $(\mathsf{pp_e}, \mathsf{pp_d}) \leftarrow \mathsf{Setup}(1^\lambda, R)$
3. $(m_0, m_1) \leftarrow \mathcal{A}(\mathsf{pp_e}, \mathsf{pp_d})$
4. $b \leftarrow \{0,1\}$
5. $c \leftarrow \mathsf{Enc}(\mathsf{pp_e}, x^*, m_b)$
6. $b' \leftarrow \mathcal{A}(c)$
7. return 1 if $(b' = b) \wedge (x^* \notin L) \wedge (|m_0| = |m_1|)$

Fig. 2: $\mathsf{Expt}_{\mathcal{A}}^{\mathsf{OWE},R}(1^\lambda)$

- $\mathsf{fk}_x \leftarrow \mathsf{PuncKey}(\mathsf{fk}, x)$: It is a deterministic algorithm that takes as input a function key fk and an element $x \in \mathcal{X}$, and produces a punctured key fk_x.
- $\mathsf{PuncF}(\mathsf{fk}_x, x') \in \mathcal{Y} \cup \{\perp\}$: It is a deterministic algorithm that takes as input a punctured key fk_x corresponding to some element $x \in \mathcal{X}$ and an element $x' \in \mathcal{X}$, and outputs a pseudorandom value $y \in \mathcal{Y}$ if $x \neq x'$. It outputs $\perp$ if $x = x'$.
- $\mathsf{Eval}(\mathsf{ek}, x, w) \in \mathcal{Y} \cup \{\perp\}$: It is a deterministic algorithm that takes as input an evaluation key ek, an element $x \in \mathcal{X}$ and a witness $w \in \mathcal{W}$, and produces an element $y \in \mathcal{Y}$ or $\perp$.

The pWPRF is said to be correct if the following properties hold:

– *correctness of* Eval: For all $x \in \mathcal{X}, w \in \mathcal{W}$ and $(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{Gen}(1^\lambda, R)$, we require that

$$\mathsf{Eval}(\mathsf{ek}, x, w) = \begin{cases} \mathsf{F}(\mathsf{fk}, x) & \text{if } R(x, w) = 1 \\ \perp & \text{if } R(x, w) = 0 \end{cases}$$

– *correctness of* PuncF: For all distinct pair of elements $x, x' \in \mathcal{X}^2$ and $(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{Gen}(1^\lambda, R)$, we require that

$$\Pr[\mathsf{F}(\mathsf{fk}, x') = \mathsf{PuncF}(\mathsf{PuncKey}(\mathsf{fk}, x), x')] = 1.$$

The security experiment $\mathsf{Expt}_{\mathcal{A}}^{\mathsf{pWPRF},R}(1^\lambda)$ for our pWPRF is defined in Fig. 1. We consider a selective model which is sufficient for our applications.

Definition 7 A puncturable witness pseudorandom function pWPRF for an NP language L with a relation R is said to be selectively secure if, for all PPT adversary $\mathcal{A}$, there exists a negligible function negl such that

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{pWPRF},R}(\lambda) = |\Pr[\mathsf{Expt}_{\mathcal{A}}^{\mathsf{pWPRF},R}(1^\lambda) = 1] - \tfrac{1}{2}| < \mathsf{negl}(\lambda)$$

In extractable pWPRF, we allow the challenge statement x^* to be in L. Accordingly, we modify the security experiment defined in Fig. 1 (in particular, line 7) and rename it as $\mathsf{Expt}_{\mathcal{A}}^{\mathsf{pWEPRF},R}(1^\lambda)$.

Definition 8 A puncturable witness pseudorandom function is said to be extractable or puncturable witness-extractable pseudorandom function (pWEPRF) for an NP language L with a relation R, if for any PPT adversary $\mathcal{A}$ there exists an extractor $\mathcal{E}$ and a polynomial $\mathsf{p}_{\mathcal{E}}$ such that, if

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{pWEPRF},R}(\lambda) = |\Pr[\mathsf{Expt}_{\mathcal{A}}^{\mathsf{pWEPRF},R}(1^\lambda) = 1] - \tfrac{1}{2}| > \alpha(\lambda)$$

for some non-negligible function α, then $\mathcal{E}(1^\lambda, x^*)$ outputs a witness $w^* \in \mathcal{W}$ such that $R(x^*, w^*) = 1$ holds with overwhelming probability and runs in time $\mathsf{p}_\mathcal{E}(\lambda, 1/\beta)$ where x^* is the challenge statement and $\beta = (\mathsf{Adv}_\mathcal{A}^{\mathsf{pWEPRF},R}(\lambda) - \alpha(\lambda))$.

2.6 Offline Witness Encryption

Definition 9 An offline witness encryption (OWE) scheme for an NP language L with a relation R is a tuple of PPT algorithms (Setup, Enc, Dec) defined as follows:

- $(\mathsf{pp_e}, \mathsf{pp_d}) \leftarrow \mathsf{Setup}(1^\lambda, R)$: It is a randomized algorithm that takes as input a security parameter λ and a relation $R : \mathcal{X} \times \mathcal{W} \rightarrow \{0, 1\}$, and produces two public parameters $\mathsf{pp_e}$ for encryption and $\mathsf{pp_d}$ for decryption.
- $c \leftarrow \mathsf{Enc}(\mathsf{pp_e}, x, m)$: It is a randomized algorithm that takes input a public parameter for encryption $\mathsf{pp_e}$, an element $x \in \mathcal{X}$ and a message $m \in \mathcal{M}$, and produces a ciphertext c.
- $\mathsf{Dec}(\mathsf{pp_d}, c, w) \in \mathcal{M} \cup \{\perp\}$: It is a deterministic algorithm that takes as input a public parameter for decryption $\mathsf{pp_d}$, a ciphertext c and a witness $w \in \mathcal{W}$, and outputs either a message $m \in \mathcal{M}$ or $\perp$.

The OWE scheme is said to be correct if the following holds:

– *correctness*: For all $x \in \mathcal{X}$, $w \in \mathcal{W}$, $m \in \mathcal{M}$ and $(\mathsf{pp_e}, \mathsf{pp_d}) \leftarrow \mathsf{Setup}(1^\lambda, R)$, we require that

$$\Pr[\mathsf{Dec}(\mathsf{pp_d}, \mathsf{Enc}(\mathsf{pp_e}, x, m), w) = m : R(x, w) = 1] = 1$$

We consider semi-adaptive security model for OWE described in the experiment $\mathsf{Expt}_\mathcal{A}^{\mathsf{OWE},R}(1^\lambda)$ (Fig. 2).

Definition 10 An offline witness encryption OWE for an NP language L with a relation R is said to be semi-adaptively secure if, for all PPT adversary $\mathcal{A}$, there exists a negligible function negl such that

$$\mathsf{Adv}_\mathcal{A}^{\mathsf{OWE},R}(\lambda) = |\Pr[\mathsf{Expt}_\mathcal{A}^{\mathsf{OWE},R}(1^\lambda) = 1] - \tfrac{1}{2}| < \mathsf{negl}(\lambda)$$

For extractable offline witness encryption we modify the experiment defined in Fig. 2 so that x^* may belong to L and rename it as $\mathsf{Expt}_\mathcal{A}^{\mathsf{EOWE},R}(1^\lambda)$.

Definition 11 An offline witness encryption OWE is said to be semi-adaptively secure extractable offline witness encryption (EOWE) for an NP language L with a relation R, if for any PPT adversary $\mathcal{A}$ there exists an extractor $\mathcal{E}$ and a polynomial $\mathsf{p}_\mathcal{E}$ such that, if

$$\mathsf{Adv}_\mathcal{A}^{\mathsf{EOWE},R}(\lambda) = |\Pr[\mathsf{Expt}_\mathcal{A}^{\mathsf{EOWE},R}(1^\lambda) = 1] - \tfrac{1}{2}| > \alpha(\lambda)$$

for some non-negligible function α, then $\mathcal{E}(1^\lambda, x^*)$ outputs a witness $w^* \in \mathcal{W}$ such that $R(x^*, w^*) = 1$ holds with overwhelming probability and runs in time $\mathsf{p}_\mathcal{E}(\lambda, 1/\beta)$ where x^* is the challenge statement and $\beta = (\mathsf{Adv}_\mathcal{A}^{\mathsf{EOWE},R}(\lambda) - \alpha(\lambda))$.

2.7 Obfuscation

Definition 12 A PPT algorithm $i\mathcal{O}$ is said to be an indistinguishability obfuscator for a class of circuits $\{\mathcal{C}_\lambda\}$, if it satisfies the following properties:

- *Functionality*: For all security parameter $\lambda \in \mathbb{N}$, for all $C \in \mathcal{C}_\lambda$, for all inputs x, we require that

$$\Pr[\widetilde{C}(x) = C(x) : \widetilde{C} \leftarrow i\mathcal{O}(1^\lambda, C)] = 1$$

- *Indistinguishability*: For any PPT distinguisher $\mathcal{D}$, there exists a negligible function negl such that for all pair of circuits $C_0, C_1 \in \mathcal{C}_\lambda$ that compute the same function and are of same size, we require that

$$\mathsf{Adv}_{\mathcal{D}}^{i\mathcal{O}}(\lambda) = |\Pr[\mathcal{D}(i\mathcal{O}(1^\lambda, C_0)) = 1] - \Pr[\mathcal{D}(i\mathcal{O}(1^\lambda, C_1)) = 1]| < \mathsf{negl}(\lambda)$$

Definition 13 A PPT algorithm $e\mathcal{O}$ is said to be an extractability obfuscator for a class of circuits $\{\mathcal{C}_\lambda\}$, if it satisfies the following properties:

- *Functionality*: For all security parameter $\lambda \in \mathbb{N}$, for all $C \in \mathcal{C}_\lambda$, for all inputs x, we require that

$$\Pr[\widetilde{C}(x) = C(x) : \widetilde{C} \leftarrow e\mathcal{O}(1^\lambda, C)] = 1$$

- *Extractability*: For any PPT distinguisher $\mathcal{D}$, there exists an extractor $\mathcal{E}$ and a polynomial $\mathsf{p}_{\mathcal{E}}$ such that for all pair of circuits $C_0, C_1 \in \mathcal{C}_\lambda$ that are of same size, for all auxiliary input $z \in \{0,1\}^*$, we require that, if

$$\mathsf{Adv}_{\mathcal{D}}^{e\mathcal{O}}(\lambda) = \big|\Pr[\mathcal{D}(e\mathcal{O}(1^\lambda, C_0), C_0, C_1, z) = 1] - \\ \Pr[\mathcal{D}(e\mathcal{O}(1^\lambda, C_1), C_0, C_1, z) = 1]\big| > \alpha(\lambda)$$

  for some non-negligible function α, then $\mathcal{E}(1^\lambda, C_0, C_1, z)$ outputs an input x such that $C_0(x) \neq C_1(x)$ holds with overwhelming probability and runs in time $\mathsf{p}_{\mathcal{E}}(\lambda, 1/\beta)$ where β is set as $(\mathsf{Adv}_{\mathcal{D}}^{e\mathcal{O}}(\lambda) - \alpha(\lambda))$.

3 Construction: (Extractable) Offline Witness Encryption

In this section, we describe our construction of $\mathsf{OWE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$ for an NP language L and a relation $R : \mathcal{X} \times \mathcal{W} \to \{0,1\}$. We consider the statement space $\mathcal{X}$ to be $\{0,1\}^\lambda$ (containing L) and $\mathcal{W} = \{0,1\}^n$ where n is a polynomial in the security parameter λ. The following primitives are utilized in our construction:

- A pseudorandom generator $\mathsf{PRG} : \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$.
- A CIND secure symmetric key encryption $\mathsf{SKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$.
- A $\mathsf{pWPRF} = (\mathsf{Gen}, \mathsf{F}, \mathsf{PuncKey}, \mathsf{PuncF}, \mathsf{Eval})$ for the NP language $L' = \{(x, v) : \exists u \in \{0,1\}^\lambda$ such that $\mathsf{PRG}(x \oplus u) = v\}$ with a relation $R' : \mathcal{X}' \times \mathcal{W}' \to \{0,1\}$. So, $R'((x, v), u) = 1$ if $\mathsf{PRG}(x \oplus u) = v$, 0 otherwise.
- An obfuscator $\mathcal{O}$ for the class of circuits $\mathcal{C}_\lambda$ required in the constructions. The only difference between the constructions of OWE and extractable OWE (EOWE) is that: $\mathcal{O}$ is an indistinguishability obfuscator ($i\mathcal{O}$) for OWE whereas $\mathcal{O}$ is an extractability obfuscator ($e\mathcal{O}$) for EOWE.

$\underline{\mathsf{Setup}(1^\lambda, R)}$:
1. $(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{pWPRF.Gen}(1^\lambda, R')$
2. $\widetilde{C} \leftarrow \mathcal{O}(1^\lambda, C[\mathsf{fk}])$
3. set $\mathsf{pp_e} = \mathsf{ek}$, $\mathsf{pp_d} = \widetilde{C}$
4. return $(\mathsf{pp_e}, \mathsf{pp_d})$

$\underline{\mathsf{Enc}(\mathsf{pp_e}, x, m)}$:
1. parse $\mathsf{pp_e} = \mathsf{ek}$
2. $u \leftarrow \{0,1\}^\lambda$, $v \leftarrow \mathsf{PRG}(x \oplus u)$
3. $y \leftarrow \mathsf{pWPRF.Eval}(\mathsf{ek}, (x, v), u)$
4. $\mathsf{K} \leftarrow \mathsf{SKE.Gen}(1^\lambda; y)$
5. $c_s \leftarrow \mathsf{SKE.Enc}(\mathsf{K}, m)$
6. return $c = (c_s, x, v)$

$\underline{C[\mathsf{fk}](c, w)}$
1. parse $c = (c_s, x, v)$
2. if $R(x, w) = 1$
3. $\quad y \leftarrow \mathsf{pWPRF.F}(\mathsf{fk}, (x, v))$
4. $\quad \mathsf{K} \leftarrow \mathsf{SKE.Gen}(1^\lambda; y)$
5. $\quad$ return $\mathsf{SKE.Dec}(\mathsf{K}, c_s)$
6. else
7. $\quad$ return $\perp$

$\underline{\mathsf{Dec}(\mathsf{pp_d}, c, w)}$:
1. parse $\mathsf{pp_d} = \widetilde{C}$
2. return $\widetilde{C}(c, w)$

Fig. 3: Construction of OWEs with optimal ciphertexts where $\mathcal{O}$ is either $i\mathcal{O}$ for normal OWE or $e\mathcal{O}$ for extractable OWE (EOWE)

Our OWE construction is shown in Fig. 3 where we assume that the circuit $C[\mathsf{fk}] \in \mathcal{C}_\lambda$ and $\mathcal{O}$ is an $i\mathcal{O}$. For correctness, we need to verify that the same key $\mathsf{K} \leftarrow \mathsf{SKE.Gen}(1^\lambda; y)$ is generated during encryption and decryption of OWE. In particular, the same randomness y should be utilized in Enc as well as in Dec. Note that, we compute y using the $\mathsf{pWPRF.Eval}(\mathsf{ek}, (x, v), \cdot)$ with a witness u corresponding to the relation R'. While decrypting, by the correctness of Eval, we generate the same y inside the circuit $\widetilde{C}$ using $\mathsf{pWPRF.F}(\mathsf{fk}, (x, v))$ extracted from the ciphertext. Therefore, $\mathsf{SKE.Dec}(\mathsf{K}, c_s)$ returns the same message that was encrypted in Enc if $R(x, w) = 1$. Finally, we conclude the correctness by observing that $C[\mathsf{fk}]$ and $\widetilde{C}$ compute the same function because of the functionality of $i\mathcal{O}$. We skip the correctness of EOWE as it can be argued similarly.
Efficiency: The ciphertext size of our OWEs is as compact as one can desire: excluding the instance, it is only $|c_s| + |v| = |m| + 2\lambda$. Note that, in SKE the size of ciphertexts are usually equal to the size of plaintexts. Hence, the ciphertext size of OWE is the size of the message added with a term proportional to the security parameter, which is fundamentally *optimal* for any public-key encryption. To encrypt a larger message, one can split the message into blocks of equal length (as supported by the SKE) and then encrypt it using a preferred modes of operation with the key K. In decryption, we use the same key K to decrypt the long ciphertext of SKE and get back the original message. The size of the public parameter for encryption ek (or $\mathsf{pp_e}$) is proportional to the size of the relation R'. We observe that the relation R' is as simple as checking a PRG computation, which means the evaluation key ek is independent of the relation R, and hence our OWE encryptions are more efficient than the reusable WE of Zhandry [16].

Theorem 1 *The* OWE = (Setup, Enc, Dec) *described in Figure 3 with* $\mathcal{O} = i\mathcal{O}$ *is a semi-adaptively secure offline witness encryption if* PRG *is a secure pseudorandom generator,* pWPRF *is a selectively secure puncturable witness pseudo-*

$x^* \leftarrow \mathcal{A}(1^\lambda)$
$(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{pWPRF.Gen}(1^\lambda, R')$
$\widetilde{C} \leftarrow i\mathcal{O}(1^\lambda, C[\mathsf{fk}])$
set $\mathsf{pp_e} = \mathsf{ek}, \mathsf{pp_d} = \widetilde{C}$
$(m_0, m_1) \leftarrow \mathcal{A}(\mathsf{pp_e}, \mathsf{pp_d})$
$u \leftarrow \{0,1\}^\lambda, v \leftarrow \mathsf{PRG}(x^* \oplus u)$
$\boxed{y \leftarrow \mathsf{pWPRF.F}(\mathsf{fk}, (x^*, v))}$
$\mathsf{K} \leftarrow \mathsf{SKE.Gen}(1^\lambda; y)$
$b \leftarrow \{0,1\}$
$c_s \leftarrow \mathsf{SKE.Enc}(\mathsf{K}, m_b)$
set $c = (c_s, x^*, v)$
$b' \leftarrow \mathcal{A}(c)$
return 1 if $(b = b')$

Fig. 4: Game 1

$x^* \leftarrow \mathcal{A}(1^\lambda)$
$(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{pWPRF.Gen}(1^\lambda, R')$
$\widetilde{C} \leftarrow i\mathcal{O}(1^\lambda, C[\mathsf{fk}])$
set $\mathsf{pp_e} = \mathsf{ek}, \mathsf{pp_d} = \widetilde{C}$
$(m_0, m_1) \leftarrow \mathcal{A}(\mathsf{pp_e}, \mathsf{pp_d})$
$\boxed{v \leftarrow \{0,1\}^{2\lambda}}$
$y \leftarrow \mathsf{pWPRF.F}(\mathsf{fk}, (x^*, v))$
$\mathsf{K} \leftarrow \mathsf{SKE.Gen}(1^\lambda; y)$
$b \leftarrow \{0,1\}$
$c_s \leftarrow \mathsf{SKE.Enc}(\mathsf{K}, m_b)$
set $c = (c_s, x^*, v)$
$b' \leftarrow \mathcal{A}(c)$
return 1 if $(b = b')$

Fig. 5: Game 2

random function, $i\mathcal{O}$ is an indistinguishability obfuscator for the circuit class $\mathcal{C}_\lambda$ and SKE *is a* CIND *secure symmetric key encryption.*

Proof. We prove the theorem using the following sequence of games. We start with Game 0 which is the standard security experiment $\mathsf{Expt}_{\mathcal{A}}^{\mathsf{OWE},R}(1^\lambda)$ as defined in Fig. 2. For Game i, we denote by $\mathsf{G_i}$ the event $b = b'$. In each game, we assume $\mathcal{A}$ submits two messages of equal length and the challenge statement $x^* \notin L$. The circuits used in the proof are assumed to be padded to a maximum size.
Game 0 $\Rightarrow$ Game 1: In Game 0, we compute the encryption key as $\mathsf{K} \leftarrow \mathsf{SKE.Gen}$ $(1^\lambda; y)$ where $y \leftarrow \mathsf{pWPRF.Eval}(\mathsf{ek}, (x^*, v), u)$. But, Game 1 (Fig. 4) sets $y \leftarrow \mathsf{pWPRF.F}(\mathsf{fk}, (x^*, v))$ without using the witness u. By the correctness Eval:

$$\mathsf{pWPRF.Eval}(\mathsf{ek}, (x^*, v), u) = \mathsf{pWPRF.F}(\mathsf{fk}, (x^*, v)) \text{ as } R'((x^*, v), u) = 1.$$

Therefore, the distribution of ciphertexts in both the games are identical and hence they are indistinguishable from $\mathcal{A}$'s view. We have $\Pr[\mathsf{G_0}] = \Pr[\mathsf{G_1}]$.

Game 1 $\Rightarrow$ Game 2: In Game 2, described in Fig. 5, we pick v uniformly at random from $\{0,1\}^{2\lambda}$ instead of setting it as $v \leftarrow \mathsf{PRG}(x^* \oplus u)$. Note that, given x^*, the distribution of $x^* \oplus u$ is uniform over $\{0,1\}^\lambda$ for $u \leftarrow \{0,1\}^\lambda$. Let, $\mathcal{B}_1$ is a PRG-adversary. Then, by the security of PRG (Def. 1), the distinguishing advantage of $\mathcal{A}$ between Game 1 and Game 2 can be written as

$$|\Pr[\mathsf{G_1}] - \Pr[\mathsf{G_2}]| = \mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{PRG}}(\lambda)$$

Game 2 $\Rightarrow$ Game 3: In Game 3, described in Fig. 6, we replace the circuit $C[\mathsf{fk}]$ by a new circuit $C[\mathsf{fk}_{z^*}, x^*]$ and set the public parameter for decryption $\mathsf{pp_d} \leftarrow i\mathcal{O}(1^\lambda, C[\mathsf{fk}_{z^*}, x^*])$. The new circuit $C[\mathsf{fk}_{z^*}, x^*]$ is defined as follows:

$C[\mathsf{fk}_{z^*}, x^*](c, w)$

parse $c = (c_s, x, v)$
if $x = x^*$
  return $\perp$
else if $R(x, w) = 1$
    $y \leftarrow \mathsf{pWPRF.PuncF}(\mathsf{fk}_{z^*}, (x, v))$

1. $x^* \leftarrow \mathcal{A}(1^\lambda)$
2. $(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{pWPRF.Gen}(1^\lambda, R')$
3. $v \leftarrow \{0,1\}^{2\lambda}$, set $z^* = (x^*, v)$
4. $\mathsf{fk}_{z^*} \leftarrow \mathsf{pWPRF.PuncKey}(\mathsf{fk}, z^*)$
5. $\boxed{\widetilde{C} \leftarrow i\mathcal{O}(1^\lambda, C[\mathsf{fk}_{z^*}, x^*])}$
6. set $\mathsf{pp_e} = \mathsf{ek}, \mathsf{pp_d} = \widetilde{C}$
7. $(m_0, m_1) \leftarrow \mathcal{A}(\mathsf{pp_e}, \mathsf{pp_d})$
8. $y \leftarrow \mathsf{pWPRF.F}(\mathsf{fk}, (x^*, v))$
9. $\mathsf{K} \leftarrow \mathsf{SKE.Gen}(1^\lambda; y)$
10. $b \leftarrow \{0,1\}$
11. $c_s \leftarrow \mathsf{SKE.Enc}(\mathsf{K}, m_b)$
12. set $c = (c_s, x^*, v)$
13. $b' \leftarrow \mathcal{A}(c)$
14. return 1 if $(b = b')$

Fig. 6: Game 3

1. $x^* \leftarrow \mathcal{A}(1^\lambda)$
2. $(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{pWPRF.Gen}(1^\lambda, R')$
3. $v \leftarrow \{0,1\}^{2\lambda}$, set $z^* = (x^*, v)$
4. $\mathsf{fk}_{z^*} \leftarrow \mathsf{pWPRF.PuncKey}(\mathsf{fk}, z^*)$
5. $\widetilde{C} \leftarrow i\mathcal{O}(1^\lambda, C[\mathsf{fk}_{z^*}, x^*])$
6. set $\mathsf{pp_e} = \mathsf{ek}, \mathsf{pp_d} = \widetilde{C}$
7. $(m_0, m_1) \leftarrow \mathcal{A}(\mathsf{pp_e}, \mathsf{pp_d})$
8. $\boxed{y \leftarrow \mathcal{Y}}$
9. $\mathsf{K} \leftarrow \mathsf{SKE.Gen}(1^\lambda; y)$
10. $b \leftarrow \{0,1\}$
11. $c_s \leftarrow \mathsf{SKE.Enc}(\mathsf{K}, m_b)$
12. set $c = (c_s, x^*, v)$
13. $b' \leftarrow \mathcal{A}(c)$
14. return 1 if $(b = b')$

Fig. 7: Game 4

```
6.         K ← SKE.Gen(1^λ; y)
7.         return SKE.Dec(K, c_s)
8.     else
9.         return ⊥
```

Note that, the two circuits $C[\mathsf{fk}]$ and $C[\mathsf{fk}_{z^*}, x^*]$ are functionally equivalent. Let $(\bar{c}, \bar{w})$ be any arbitrary input where $\bar{c} = (\bar{c}_s, \bar{x}, \bar{v})$. If $\bar{x} = x^*$, then $C[\mathsf{fk}](\bar{c}, \bar{w})$ outputs $\perp$ since $x^* \notin L$ implies that $R(x^*, \bar{w}) = 0$ for any $\bar{w} \in \mathcal{W}$, and $C[\mathsf{fk}_{z^*}, x^*](\bar{c}, \bar{w})$ outputs $\perp$ because of the check in line 2 of the circuit. If $\bar{x} \neq x^*$, then $z^* \neq (\bar{x}, \bar{v})$ and by the correctness of PuncF we have

$$\mathsf{pWPRF.F}(\mathsf{fk}, (\bar{x}, \bar{v})) = \mathsf{pWPRF.PuncF}(\mathsf{fk}_{z^*}, (\bar{x}, \bar{v}))$$

and hence $C[\mathsf{fk}](\bar{c}, \bar{w}) = C[\mathsf{fk}_{z^*}, x^*](\bar{c}, \bar{w})$. Considering $\mathcal{D}$ as a PPT distinguisher for $i\mathcal{O}$, the indistinguishability property of $i\mathcal{O}$ (Def. 12) implies that

$$|\Pr[\mathsf{G}_2] - \Pr[\mathsf{G}_3]| = \mathsf{Adv}_{\mathcal{D}}^{i\mathcal{O}}(\lambda)$$

<u>Game 3 $\Rightarrow$ Game 4</u>: In Game 4, described in Fig. 7, we pick y uniformly at random from $\mathcal{Y}$ which is the co-domain of $\mathsf{pWPRF.F}(\mathsf{fk}, \cdot)$. We show that if $\mathcal{A}$ can distinguish between these two games, then there is an adversary $\mathcal{B}_2$ which will break the selective security of pWPRF (defined in Fig. 1). Let $z^* = (x^*, v)$ be the challenge statement of $\mathcal{B}_2$ for a random $v \leftarrow \{0,1\}^{2\lambda}$.

<u>$\mathcal{B}_2(1^\lambda, z^*)$</u>:

1. send z^* to its challenger
2. The pWPRF-challenger does the following:
   (a) generate $(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{pWPRF.Gen}(1^\lambda, R')$
   (b) compute a punctured key $\mathsf{fk}_{z^*} \leftarrow \mathsf{pWPRF.PuncKey}(\mathsf{fk}, z^*)$
   (c) set $y_0 \leftarrow \mathsf{pWPRF.F}(\mathsf{fk}, z^*)$ and $y_1 \leftarrow \mathcal{Y}$
   (d) pick $\tilde{b} \leftarrow \{0,1\}$
   (e) return $(\mathsf{ek}, \mathsf{fk}_{z^*}, y_{\tilde{b}})$ to $\mathcal{B}_2$
3. compute $\widetilde{C} \leftarrow i\mathcal{O}(1^\lambda, C[\mathsf{fk}_{z^*}, x^*])$ and set $\mathsf{pp_e} = \mathsf{ek}, \mathsf{pp_d} = \widetilde{C}$
4. receive $(m_0, m_1) \leftarrow \mathcal{A}(\mathsf{pp_e}, \mathsf{pp_d})$
5. compute the encryption key as $\mathsf{K} \leftarrow \mathsf{SKE.Gen}(1^\lambda; y_{\tilde{b}})$

6. pick $b \leftarrow \{0,1\}$
7. compute the ciphertext as $c_s \leftarrow \mathsf{SKE.Enc}(\mathsf{K}, m_b)$
8. set $c = (c_s, x^*, v)$
9. get $b' \leftarrow \mathcal{A}(c)$
10. return 1 if $(b = b')$

First, we note that $z^* = (x^*, v) \notin L'$ with overwhelming probability. Since $v \leftarrow \{0,1\}^{2\lambda}$, the probability that $\mathsf{PRG}(x^* \oplus u) = v$ for some $u \in \{0,1\}^\lambda$ is at most $2^{-\lambda}$ which is negligible in λ. So, $\mathcal{B}_2$ is a legitimate pWPRF-adversary. If the pWPRF-challenger picks $\tilde{b} = 0$ then $\mathcal{B}_2$ simulates $\mathsf{Game\ 3}$, and if it chooses $\tilde{b} = 1$ then $\mathcal{B}_2$ simulates $\mathsf{Game\ 4}$. Therefore, the advantage of $\mathcal{A}$ in distinguishing between $\mathsf{Game\ 3}$ and $\mathsf{Game\ 4}$ is the same as the advantage of $\mathcal{B}_2$ in breaking the selective security of pWPRF. Hence the following holds:

$$|\Pr[\mathsf{G}_3] - \Pr[\mathsf{G}_4]| = \mathsf{Adv}_{\mathcal{B}_2}^{\mathsf{pWPRF},R'}(\lambda)$$

Next, we note that in $\mathsf{Game\ 4}$, the encryption key is computed as $\mathsf{K} \leftarrow \mathsf{SKE.Gen}(1^\lambda; y)$ with a fresh randomness y which is independent of the challenge statement x^*. Therefore, by the CIND security of SKE (Def. 5) we have

$$|\Pr[\mathsf{G}_4] - \tfrac{1}{2}| = \mathsf{Adv}_{\mathcal{B}_3}^{\mathsf{SKE}}(\lambda)$$

where $\mathcal{B}_3$ is an adversary of CIND security game. Finally, we conclude the proof by combining all the probabilities as follows:

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{OWE},R}(\lambda) = |\Pr[\mathsf{G}_0] - \frac{1}{2}| &\leq \sum_{i=0}^{3} |\Pr[\mathsf{G}_i] - \Pr[\mathsf{G}_{i+1}]| + |\Pr[\mathsf{G}_4] - \frac{1}{2}| \\ &= \mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{PRG}}(\lambda) + \mathsf{Adv}_{\mathcal{D}}^{i\mathsf{O}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_2}^{\mathsf{pWPRF},R'}(\lambda) + \mathsf{Adv}_{\mathcal{B}_3}^{\mathsf{SKE}}(\lambda) \\ &< \mathsf{negl}(\lambda) \quad \text{(by the assumptions in the theorem)}\end{aligned}$$

In the next theorem, we proof the security of EOWE (Fig. 3 with $\mathcal{O} = e\mathcal{O}$) utilizing the extractor of $e\mathcal{O}$.

Theorem 2 *The* $\mathsf{EOWE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$ *described in Figure 3 with* $\mathcal{O} = e\mathcal{O}$ *is a semi-adaptively secure extractable offline witness encryption if* PRG *is a secure pseudorandom generator,* pWPRF *is a selectively secure puncturable witness pseudorandom function,* $e\mathcal{O}$ *is an extractability obfuscator for the circuit class* $\mathcal{C}_\lambda$ *and* SKE *is a* CIND *secure symmetric key encryption.*

Proof. We start with the standard EOWE experiment $\mathsf{Expt}_{\mathcal{A}}^{\mathsf{EOWE},R}(1^\lambda)$ (Def. 11). We call it as $\mathsf{EGame\ 0}$. Here, we denote the security games by $\mathsf{EGame\ i}$ and for each $\mathsf{EGame\ i}$, let $\mathsf{EG_i}$ be the event $b = b'$. We assume that $\mathcal{A}$ submits two messages of equal length in each game and all the circuits used in the proof are padded to a maximum size.

$\underline{\mathsf{EGame\ 0} \Rightarrow \mathsf{EGame\ 1}}$: $\mathsf{EGame\ 1}$ is exactly the same as $\mathsf{EGame\ 0}$ except we replace the circuit $C[\mathsf{fk}]$ with a new circuit $C[\mathsf{fk}, x^*]$ defined in Fig. 8. Suppose, the adversary $\mathcal{A}$ can distinguish between $\mathsf{EGame\ 0}$ and $\mathsf{EGame\ 1}$ with an advantage

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{EGame\ 0\text{-}1}}(\lambda) = |\Pr[\mathsf{EG}_0] - \Pr[\mathsf{EG}_1]| > \alpha(\lambda)$$

1. $x^* \leftarrow \mathcal{A}(1^\lambda)$
2. $(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{pWPRF.Gen}(1^\lambda, R')$
3. $\boxed{\widetilde{C} \leftarrow e\mathcal{O}(1^\lambda, C[\mathsf{fk}, x^*])}$
4. set $\mathsf{pp_e} = \mathsf{ek}, \mathsf{pp_d} = \widetilde{C}$
5. $(m_0, m_1) \leftarrow \mathcal{A}(\mathsf{pp_e}, \mathsf{pp_d})$
6. $u \leftarrow \{0,1\}^\lambda, v \leftarrow \mathsf{PRG}(x^* \oplus u)$
7. $y \leftarrow \mathsf{pWPRF.Eval}(\mathsf{ek}, (x^*, v), u)$
8. $\mathsf{K} \leftarrow \mathsf{SKE.Gen}(1^\lambda; y)$
9. $b \leftarrow \{0,1\}$
10. $c_s \leftarrow \mathsf{SKE.Enc}(\mathsf{K}, m_b)$
11. set $c = (c_s, x^*, v)$
12. $b' \leftarrow \mathcal{A}(c)$
13. return 1 if $b = b'$

$\underline{C[\mathsf{fk}, x^*](c, w)}$

1. parse $c = (c_s, x, v)$
2. if $R(x, w) = 1$
3.   if $x = x^*$
4.   return $\perp$
5.   else
6. $y \leftarrow \mathsf{pWPRF.F}(\mathsf{fk}, (x, v))$
7. $\mathsf{K} \leftarrow \mathsf{SKE.Gen}(1^\lambda; y)$
8.   return $\mathsf{SKE.Dec}(\mathsf{K}, c_s)$
9. else
10.   return $\perp$

Fig. 8: EGame 1

for some non-negligible function α. Then, we show that there is an extractor $\mathcal{E}$ and a polynomial $\mathsf{p}_\mathcal{E}$ such that $\mathcal{E}(1^\lambda, x^*)$ outputs a witness w^* satisfying $R(x^*, w^*) = 1$ with overwhelming probability and runs in time $\mathsf{p}_\mathcal{E}(\lambda, 1/\beta)$ where $\beta = (\mathsf{Adv}_\mathcal{A}^{\mathsf{EGame\ 0\text{-}1}}(\lambda) - \alpha(\lambda))$.

We note that two games differ only in the obfuscated circuits. So, we consider a PPT distinguisher $\mathcal{D}$ of $e\mathcal{O}$ as defined in Def. 13. In particular, $\mathcal{D}$ collects two circuits from a circuit sampler $\mathsf{S}(1^\lambda, \cdot)$ and an obfuscated circuit (from it's challenger), then it simulates the security game for $\mathcal{A}$ as follows:

$\underline{\mathcal{D}(1^\lambda, \widetilde{C}, C[\mathsf{fk}], C[\mathsf{fk}, x^*], \mathsf{aux})}$:

1. parse $\mathsf{aux} = (\mathsf{ek}, x^*)$
2. set $\mathsf{pp_e} = \mathsf{ek}, \mathsf{pp_d} = \widetilde{C}$
3. $(m_0, m_1) \leftarrow \mathcal{A}(\mathsf{pp_e}, \mathsf{pp_d})$
4. follow steps 6-10 as in EGame 1
5. set $c = (c_s, x^*, v)$
6. $b' \leftarrow \mathcal{A}(c)$
7. return 1 if $b = b'$

$\underline{\mathsf{S}(1^\lambda, x^*)}$

1. $(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{pWPRF.Gen}(1^\lambda, R')$
2. construct $C[\mathsf{fk}], C[\mathsf{fk}, x^*]$
3. set $\mathsf{aux} = (\mathsf{ek}, x^*)$
4. return $(C[\mathsf{fk}], C[\mathsf{fk}, x^*], \mathsf{aux})$

If $\widetilde{C} \leftarrow e\mathcal{O}(1^\lambda, C[\mathsf{fk}])$ then $\mathcal{D}$ simulates EGame 0 and if $\widetilde{C} \leftarrow e\mathcal{O}(1^\lambda, C[\mathsf{fk}, x^*])$ then $\mathcal{D}$ simulates EGame 1. Therefore, $\mathcal{D}$ can distinguish between the obfuscated circuits with the same advantage of $\mathcal{A}$ in distinguishing EGame 0 and EGame 1. By the extractability property of $e\mathcal{O}$ (Def. 13), there exists an extractor $\mathcal{E}'$ and a polynomial $\mathsf{p}_{\mathcal{E}'}$ such that $\mathcal{E}'(1^\lambda, C[\mathsf{fk}], C[\mathsf{fk}, x^*], \mathsf{aux})$ outputs $(\bar{c}, \bar{w})$ at which the two circuits differ and runs in time $\mathsf{p}_{\mathcal{E}'}(\lambda, 1/\beta)$ with $\beta = (\mathsf{Adv}_\mathcal{A}^{\mathsf{EGame\ 0\text{-}1}}(\lambda) - \alpha(\lambda))$. Note that, the two circuits differ only when $\bar{c} = (\bar{c}_s, x^*, \bar{v})$ is well formed and $R(x^*, \bar{w}) = 1$.

Now, the extractor $\mathcal{E}(1^\lambda, x^*)$ of EOWE simply runs $\mathsf{S}(1^\lambda, x^*)$ to obtain $(C[\mathsf{fk}], C[\mathsf{fk}, x^*], \mathsf{aux})$ and then executes $\mathcal{E}'(1^\lambda, C[\mathsf{fk}], C[\mathsf{fk}, x^*], \mathsf{aux})$ to get a witness w^* satisfying $R(x^*, w^*) = 1$ with high probability. The runtime of $\mathcal{E}$ is equal to the runtime of S plus the runtime of $\mathcal{E}'$, hence is bounded by $\text{poly}(\lambda) + \mathsf{p}_{\mathcal{E}'}(\lambda, 1/\beta) = \mathsf{p}_\mathcal{E}(\lambda, 1/\beta)$ for some polynomial $\mathsf{p}_\mathcal{E}$ where $\beta = (\mathsf{Adv}_\mathcal{A}^{\mathsf{EGame\ 0\text{-}1}}(\lambda) - \alpha(\lambda))$.

$\underline{\mathsf{EGame\ 1} \Rightarrow \mathsf{EGame\ 2}}$: EGame 2 is exactly the same as EGame 1 except in line 7 of Fig. 8 where we compute $y \leftarrow \mathsf{pWPRF.F}(\mathsf{fk}, (x^*, v))$. By the correctness Eval (using the same argument as in the transition from $\mathsf{Game\ 0}$ to $\mathsf{Game\ 1}$ of Th. 1), we have $\Pr[\mathsf{EG}_1] = \Pr[\mathsf{EG}_2]$.

<u>$\mathsf{EGame\ 2} \Rightarrow \mathsf{EGame\ 3}$</u>: In $\mathsf{EGame\ 3}$, we choose $v \leftarrow \{0,1\}^{2\lambda}$ instead of computing $v \leftarrow \mathsf{PRG}(x^* \oplus u)$ as in $\mathsf{EGame\ 2}$. By the security of PRG (Def. 1), we have

$$|\Pr[\mathsf{EG}_2] - \Pr[\mathsf{EG}_3]| = \mathsf{Adv}^{\mathsf{PRG}}_{\mathcal{B}_1}(\lambda)$$

where $\mathcal{B}_1$ is a PRG-adversary.

<u>$\mathsf{EGame\ 3} \Rightarrow \mathsf{EGame\ 4}$</u>: In $\mathsf{EGame\ 4}$, we set $\mathsf{pp_d} \leftarrow e\mathcal{O}(1^\lambda, C[\mathsf{fk}_{z^*}, x^*])$ where $\mathsf{fk}_{z^*} \leftarrow \mathsf{pWPRF.PuncKey}(\mathsf{fk}, z^*)$ and $z^* = (x^*, v)$ for some $v \leftarrow \{0,1\}^{2\lambda}$. The circuit $C[\mathsf{fk}_{z^*}, x^*]$ is the same circuit defined in Fig. 8 except we replace fk by fk_{z^*} and use $\mathsf{pWPRF.PuncF}(\mathsf{fk}_{z^*}, (x, v))$ to compute y in line 6. It is easy to follow that the circuits $C[\mathsf{fk}, x^*], C[\mathsf{fk}_{z^*}, x^*]$ compute the same function by the correctness of PuncF. Suppose, $(\bar{c} = (\bar{c}_s, \bar{x}, \bar{v}), \bar{w})$ is any arbitrary input to the circuits. If $\bar{x} \neq x^*$, then $z^* \neq (\bar{x}, \bar{v})$ and hence $\mathsf{pWPRF.F}(\mathsf{fk}, (\bar{x}, \bar{v})) = \mathsf{pWPRF.PuncF}(\mathsf{fk}_{z^*}, (\bar{x}, \bar{v}))$. If $\bar{x} = x^*$, then both the circuits return $\perp$ because of the check in line 2 or 3. By the extractability property of $e\mathcal{O}$ (Def. 13), we have

$$|\Pr[\mathsf{EG}_3] - \Pr[\mathsf{EG}_4]| = \mathsf{Adv}^{e\mathcal{O}}_{\mathcal{D}}(\lambda) = \mu(\lambda)$$

where μ is a negligible function of λ. If the advantage is not bounded by a negligible function of λ, then there exists an extractor $\mathcal{E}'$ which would produce an input where the two circuits differ, leading towards a contradiction as the circuits are equivalent.

<u>$\mathsf{EGame\ 4} \Rightarrow \mathsf{EGame\ 5}$</u>: $\mathsf{EGame\ 5}$ samples y uniformly at random from $\mathcal{Y}$ instead of computing $y \leftarrow \mathsf{pWPRF.F}(\mathsf{fk}, (x^*, v))$ as in $\mathsf{EGame\ 4}$, where $\mathcal{Y}$ is the co-domain of $\mathsf{pWPRF.F}(\mathsf{fk}, \cdot)$. Note that the probability of $z^* = (x^*, v) \in L'$ for a random $v \leftarrow \{0,1\}^{2\lambda}$ is negligible in λ. By the selective security of pWPRF, we have

$$|\Pr[\mathsf{EG}_4] - \Pr[\mathsf{EG}_5]| = \mathsf{Adv}^{\mathsf{pWPRF}, R'}_{\mathcal{B}_2}(\lambda)$$

where $\mathcal{B}_2$ is a pWPRF-adversary. We skip the reduction as it is similar to the reduction described in the transition from $\mathsf{Game\ 3}$ to $\mathsf{Game\ 4}$ of Th. 1.

Finally, the encryption key in $\mathsf{EGame\ 5}$ is computed as $\mathsf{K} \leftarrow \mathsf{SKE.Gen}(1^\lambda; y)$ where y is a fresh randomness which is independent of the challenge statement x^*. The CIND security of SKE (Def. 5) guarantees that

$$|\Pr[\mathsf{EG}_5] - \tfrac{1}{2}| = \mathsf{Adv}^{\mathsf{SKE}}_{\mathcal{B}_3}(\lambda).$$

where $\mathcal{B}_3$ is an adversary of CIND game. Combining all the probabilities, we have

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{EOWE}, R}_{\mathcal{A}}(\lambda) = |\Pr[\mathsf{EG}_0] - \frac{1}{2}| &\leq \sum_{i=0}^{4} |\Pr[\mathsf{EG}_i] - \Pr[\mathsf{EG}_{i+1}]| + |\Pr[\mathsf{EG}_5] - \frac{1}{2}| \\
&= \mathsf{Adv}^{\mathsf{EGame\ 0\text{-}1}}_{\mathcal{A}}(\lambda) + \mathsf{Adv}^{\mathsf{PRG}}_{\mathcal{B}_1}(\lambda) + \mu(\lambda) \\
&\quad + \mathsf{Adv}^{\mathsf{pWPRF}, R'}_{\mathcal{B}_2}(\lambda) + \mathsf{Adv}^{\mathsf{SKE}}_{\mathcal{B}_3}(\lambda) \\
&< \mathsf{Adv}^{\mathsf{EGame\ 0\text{-}1}}_{\mathcal{A}}(\lambda) + \mathsf{negl}(\lambda) \quad \text{(by the assumptions in the theorem)}
\end{aligned}$$

Thus, $|\mathsf{Adv}_{\mathcal{A}}^{\mathsf{EOWE},R}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{\mathsf{EGame\ 0\text{-}1}}(\lambda)| < \mathsf{negl}(\lambda)$ implies $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{EGame\ 0\text{-}1}}(\lambda) = \mathsf{Adv}_{\mathcal{A}}^{\mathsf{EOWE},R}(\lambda)$ excluding the negligible term. Hence, by the similar arguments as in the transition from EGame 0 to EGame 1, we conclude that if $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{EOWE},R}(\lambda) > \varrho(\lambda)$ for some non-negligible function ϱ, then there is an extractor $\mathcal{E}$ and a polynomial $\mathsf{p}_{\mathcal{E}}$ such that $\mathcal{E}(1^{\lambda}, x^*)$ outputs a witness w^* satisfying $R(x^*, w^*) = 1$ with overwhelming probability and runs in time $\mathsf{p}_{\mathcal{E}}(\lambda, 1/\beta)$ where $\beta = (\mathsf{Adv}_{\mathcal{A}}^{\mathsf{EOWE},R}(\lambda) - \varrho(\lambda))$. This completes the proof.

4 Informal Description: (Extractable) Offline Functional Witness Encryption

Apart from an NP language L with a witness relation R, *Offline functional witness encryption* (OFWE) is associated with a function class $\{\mathcal{F}_\lambda\}$. It encrypts a pair of function and message $(f, m) \in \mathcal{F}_\lambda \times \mathcal{M}$ with respect to a statement x. Instead of getting the whole message, a valid witness w for the statement x can only get a user to learn $f(m, w)$. The OWE described in Fig. 3 can be modified to achieve OFWE. While encryption, we use the key K (computed utilizing pWPRF.Eval for the statement (x, v)) to encrypt (f, m) via SKE encryption. The ciphertext becomes $c = (c_s, x, v)$ with $|c_s| = |f| + |m|$ where $|f|, |m|$ denote the sizes of f, m respectively. In Setup, we modify $C[\mathsf{fk}]$ in line 5 so that the circuit computes $(f, m) \leftarrow \mathsf{SKE.Dec}(\mathsf{K}, c_s)$ and then returns $f(m, w)$ if $R(x, w) = 1$ holds. The rest of the construction remains the same. Note that the size of ciphertext is optimal and the encryption maintains similar efficiency akin to our OWE. For security, we consider semi-adaptive model where the adversary $\mathcal{A}$ commits on the challenge statement x^* before the setup and adaptively selects two pairs (f_0, m_0), (f_1, m_1) such that $f_0(m_0, w) = f_1(m_1, w)$ for all w satisfying $R(x^*, w) = 1$. Detail construction with security (Th. 5) is described in App. C.

Replacing $i\mathcal{O}$ with an $e\mathcal{O}$ leads us to an *extractable* OFWE which is selectively secure means that $\mathcal{A}$ submits a challenge tuple (x^*, f, m_0, m_1) before setup. Depending on the wining advantage of $\mathcal{A}$ in guessing the bit b hidden inside a ciphertext corresponding to (x^*, f, m_b), there exists an extractor $\mathcal{E}$ which on input the challenge tuple outputs a witness w satisfying $f(m_0, w) \neq f(m_1, w)$ and $R(x^*, w) = 1$ with high probability. We prove the security in Th. 6, App. C.

5 Construction: Puncturable Witness(-Extractable) Pseudorandom Function

In this section, we show that WPRF construction of [14] satisfies our definition of pWPRF. In addition, we observe that if the indistinguishability obfuscator is replaced with an extractability obfuscator then the pWPRF becomes extractable. We now describe the pWPRF = (Gen, F, PuncKey, PuncF, Eval) for any NP language L with a relation $R : \mathcal{X} \times \mathcal{W} \to \{0, 1\}$. The following primitives are required for the construction.

– A pPRF = (Gen, PuncKey, Eval, PuncEval) with domain $\mathcal{X}$ and co-domain $\mathcal{Y}$.

$\underline{\mathsf{Gen}(1^\lambda, R)}$:

1. $\mathsf{K} \leftarrow \mathsf{pPRF.Gen}(1^\lambda)$
2. $\widetilde{C} \leftarrow \mathcal{O}(1^\lambda, C[\mathsf{K}])$
3. set $\mathsf{fk} = \mathsf{K}$, $\mathsf{ek} = \widetilde{C}$
4. return $(\mathsf{fk}, \mathsf{ek})$

$\underline{\mathsf{pWPRF.F}(\mathsf{fk}, x)}$:

1. parse $\mathsf{fk} = \mathsf{K}$
2. set $y \leftarrow \mathsf{pPRF.Eval}(\mathsf{K},\ x)$
3. return y

$\underline{\mathsf{pWPRF.PuncKey}(\mathsf{fk}, x)}$:

1. parse $\mathsf{fk} = \mathsf{K}$
2. set $\mathsf{fk}_x \leftarrow \mathsf{pPRF.PuncKey}(\mathsf{K},\ x)$
3. return fk_x

$\underline{C[\mathsf{K}](x, w)}$

1. if $R(x, w) = 1$
2. set $y \leftarrow \mathsf{pPRF.Eval}(\mathsf{K}, x)$
3. return y
4. else
5. return $\perp$

$\underline{\mathsf{pWPRF.PuncF}(\mathsf{fk}_x, x')}$

1. return $\mathsf{pPRF.PuncEval}(\mathsf{fk}_x, x')$

$\underline{\mathsf{pWPRF.Eval}(\mathsf{ek}, x, w)}$:

1. parse $\mathsf{ek} = \widetilde{C}$
2. return $\widetilde{C}(x, w)$

Fig. 9: Construction of pWPRFs where $\mathcal{O}$ is either $i\mathcal{O}$ for normal pWPRF or $e\mathcal{O}$ for extractable pWPRF (pWEPRF)

– An obfuscator $\mathcal{O}$ for the class of circuits $\mathcal{C}_\lambda$ required in the constructions. The only difference between the constructions of pWPRF and pWEPRF is that: $\mathcal{O}$ is an indistinguishability obfuscator ($i\mathcal{O}$) for pWPRF whereas $\mathcal{O}$ is an extractability obfuscator ($e\mathcal{O}$) for pWEPRF.

The constructions of pWPRFs are shown in Fig. 9. The correctness directly follows from the correctness of the underlying pPRF and functionality of $\mathcal{O}$.

Theorem 3 *The* pWEPRF = (Gen, F, PuncKey, PuncF, Eval) *described in Figure 5 with* $\mathcal{O} = i\mathcal{O}$ *is a selectively secure puncturable witness pseudorandom function if* pPRF *is a secure puncturable pseudorandom function and* $i\mathcal{O}$ *is an indistinguishability obfuscator for the circuit class* $\mathcal{C}_\lambda$*.*

Proof sketch. As usual, we start with game 0 which is the standard security experiment $\mathsf{Expt}_{\mathcal{A}}^{\mathsf{pWPRF},R}(1^\lambda)$ as defined in Fig. 1. Next, in game 1, we replace the circuit $C[\mathsf{K}]$ with a new circuit $C[\mathsf{fk}_{x^*}, x^*]$ where $\mathsf{fk}_{x^*} \leftarrow \mathsf{pPRF.PuncKey}(\mathsf{K}, x^*)$. For any arbitrary input (x, w), the new circuit returns the pseudorandom value as $\mathsf{pPRF.PuncEval}(\mathsf{fk}_{x^*}, x)$ if $x \neq x^*$ and $R(x, w) = 1$ hold, otherwise it returns $\perp$. It is easy to verify that the two circuits are functionally equivalent and hence by the security of $i\mathcal{O}$, game 0 and game 1 are indistinguishable. Now, the adversary knowing fk_{x^*} cannot distinguish $\mathsf{pWPRF.F}(\mathsf{fk},\ x^*)$ from a random element due to the security of underlying pPRF (Def. 3). A formal proof is given in App. A.

We discuss the security of pWEPRF in App. B where the extractibility property of obfuscation (Def. 13) is utilized.

6 Conclusion

In this paper, we initiate the study of puncturable WPRF(pWPRF). We demonstrate that this puncturing technique enhances the applicability of pWPRF. We achieve optimal size ciphertext for OWE with semi-adaptive security. Additionally, we observe that OWE encryption is independent of the original relation which means the encryption efficiency does not rely on the NP language. We further see that the OWE can be extended to offline functional WE (OFWE) providing more control over data. Moreover, using $e\mathcal{O}$ we construct extractable OWE and extractable OFWE with similar efficiency of encryption.

In future, we expect more cryptographic primitives realized from pWPRF. In terms of security, it is desirable to construct WPRF in adaptive model without multilinear maps [16]. This may lead us to OWE with full adaptive security. Finally, we note that a significant open problem in this area is to construct WPRF or OWE based on standard assumptions related to bilinear maps or lattices.

References

1. H. Abusalah, G. Fuchsbauer, and K. Pietrzak. Offline witness encryption. In *International Conference on Applied Cryptography and Network Security*, pages 285–303. Springer, 2016.
2. S. Agrawal. Indistinguishability obfuscation without multilinear maps: new methods for bootstrapping and instantiation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 191–225. Springer, 2019.
3. P. Ananth, A. Jain, H. Lin, C. Matt, and A. Sahai. Indistinguishability obfuscation without multilinear maps: New paradigms via low degree weak pseudorandomness and security amplification. Cryptology ePrint Archive, Report 2019/643, 2019. `https://eprint.iacr.org/2019/643`.
4. M. Barbosa, B. Portela, G. Scerri, and B. Warinschi. Foundations of hardware-based attested computation and application to sgx. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 245–260. IEEE, 2016.
5. M. Bellare and V. T. Hoang. Adaptive witness encryption and asymmetric password-based cryptography. In *IACR International Workshop on Public Key Cryptography*, pages 308–331. Springer, 2015.
6. E. Boyle, K.-M. Chung, and R. Pass. On extractability (aka differing-inputs) obfuscation. TCC, 2014.
7. J. H. Cheon, W. Cho, M. Hhan, J. Kim, and C. Lee. Statistical zeroizing attack: Cryptanalysis of candidates of bp obfuscation over ggh15 multilinear map. Cryptology ePrint Archive, Report 2018/1081, 2018. `https://eprint.iacr.org/2018/1081`.
8. P. Chvojka, T. Jager, and S. A. Kakvi. Offline witness encryption with semi-adaptive security. Cryptology ePrint Archive, Report 2019/1337, 2019. `https://eprint.iacr.org/2019/1337`.
9. J.-S. Coron and L. Notarnicola. Cryptanalysis of clt13 multilinear maps with independent slots. *IACR Cryptology ePrint Archive*, 2019:309, 2019.
10. B. Fisch, D. Vinayagamurthy, D. Boneh, and S. Gorbunov. Iron: functional encryption using intel sgx. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 765–782. ACM, 2017.
11. S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. *SIAM Journal on Computing*, 45(3):882–929, 2016.
12. S. Garg, C. Gentry, A. Sahai, and B. Waters. Witness encryption and its applications. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 467–476. ACM, 2013.
13. J. Liu, T. Jager, S. A. Kakvi, and B. Warinschi. How to build time-lock encryption. *Designs, Codes and Cryptography*, 86(11):2549–2586, 2018.
14. T. Pal and R. Dutta. Offline witness encryption from witness prf and randomized encoding in crs model. In *Australasian Conference on Information Security and Privacy*, pages 78–96. Springer, 2019.
15. A. Sahai and B. Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 475–484. ACM, 2014.
16. M. Zhandry. How to avoid obfuscation using witness prfs. In *Theory of Cryptography Conference*, pages 421–448. Springer, 2016.

A A Formal Proof of Theorem 3

Proof. We prove the security using two games. We start with $\mathsf{Game\ 0}$ which is the standard selective security experiment as in Def. 6. Let $\mathsf{G_i}$ be the event $b = b'$ in each $\mathsf{Game\ i}$.

<u>$\mathsf{Game\ 0} \Rightarrow \mathsf{Game\ 1}$</u>: $\mathsf{Game\ 1}$ is exactly same as the $\mathsf{Game\ 0}$ except we replace the circuit $C[\mathsf{K}]$ with a new circuit $C[\mathsf{fk}_{x^*}, x^*]$ defined in Fig. 10, where $\mathsf{fk}_{x^*} \leftarrow \mathsf{pPRF.PuncKey}(\mathsf{K}, x^*)$. We show that the two circuits $C[\mathsf{K}]$ and $C[\mathsf{fk}_{x^*}, x^*]$ are functionally equivalent. For any arbitrary input $(\bar{x}, \bar{w})$ to the circuits, we see that if $\bar{x} \neq x^*$, then both the circuits return the same value as $\mathsf{pPRF.Eval}(\mathsf{K}, \bar{x}) = \mathsf{pPRF.PuncEval}(\mathsf{fk}_{x^*}, \bar{x})$. Otherwise, if $\bar{x} = x^*$ then the circuit $C[\mathsf{K}]$ returns $\perp$, because $x^* \notin L$ implies that $R(\bar{x}, \bar{w}) = 0$ for all $\bar{w} \in \mathcal{W}$, and the circuit $C[\mathsf{fk}_{x^*}, x^*]$ returns $\perp$ because of the check in line 2 (Fig. 10). Thus, the indistinguishability property of $i\mathcal{O}$ (Def. 12) guarantees that

$$|\Pr[\mathsf{G}_0] - \Pr[\mathsf{G}_1]| = \mathsf{Adv}_{\mathcal{D}}^{i\mathcal{O}}(\lambda)$$

where $\mathcal{D}$ is a PPT distinguisher for $i\mathcal{O}$.

1. $x^* \leftarrow \mathcal{A}(1^\lambda)$
2. $\mathsf{K} \leftarrow \mathsf{pPRF.Gen}(1^\lambda)$
3. $\boxed{\widetilde{C} \leftarrow i\mathcal{O}(1^\lambda, C[\mathsf{fk}_{x^*}, x^*])}$
4. set $\mathsf{ek} = \widetilde{C}$
5. $\mathsf{fk}_{x^*} \leftarrow \mathsf{pPRF.PuncKey}(\mathsf{K}, x^*)$
6. $y_0 \leftarrow \mathsf{pPRF.Eval}(\mathsf{K}, x^*)$, $y_1 \leftarrow \mathcal{Y}$
7. $b \leftarrow \{0, 1\}$
8. $b' \leftarrow \mathcal{A}(\mathsf{ek}, \mathsf{fk}_{x^*}, y_b)$
9. return 1 if $b = b'$

<u>$C[\mathsf{fk}_{x^*}, x^*](x, w)$</u>

```
1. if R(x, w) = 1
2.   if x = x*
3.     return ⊥
4.   else
5.     y ← pPRF.PuncEval(fk_{x*}, x)
6.     return y
7. else
8.   return ⊥
```

Fig. 10: $\mathsf{Game\ 1}$

Suppose, the advantage of $\mathcal{A}$ in $\mathsf{Game\ 1}$ is non-negligible. Then we construct an adversary $\mathcal{B}$ against the security of pPRF (Def. 2) with the same advantage as follow.

<u>$\mathcal{B}(1^\lambda, x^*)$</u>:

1. send x^* to its challenger
2. The pPRF-challenger does the following:
   (a) generate $\mathsf{K} \leftarrow \mathsf{pPRF.Gen}(1^\lambda)$
   (b) compute $\mathsf{fk}_{x^*} \leftarrow \mathsf{pPRF.PuncKey}(\mathsf{K}, x^*)$
   (c) set $y_0 \leftarrow \mathsf{pPRF.Eval}(\mathsf{K}, x^*)$ and $y_1 \leftarrow \mathcal{Y}$
   (d) pick $b \leftarrow \{0, 1\}$
   (e) return (fk_{x^*}, y_b) to $\mathcal{B}$
3. compute $\widetilde{C} \leftarrow i\mathcal{O}(1^\lambda, C[\mathsf{fk}_{x^*}, x^*])$ and set $\mathsf{ek} = \widetilde{C}$
4. get $b' \leftarrow \mathcal{A}(\mathsf{ek}, \mathsf{fk}_{x^*}, y_b)$
5. return 1 if $b = b'$

Note that $\mathcal{B}$ perfectly simulates Game 1 for $\mathcal{A}$. If $\mathcal{A}$ can guess the bit b in Game 1 with a non-negligible advantage, then $\mathcal{B}$ breaks the security of pPRF with the same advantage. From the security of pPRF, we have

$$|\Pr[\mathsf{G}_1] - \tfrac{1}{2}| = \mathsf{Adv}_{\mathcal{B}}^{\mathsf{pPRF}}(\lambda)$$

Combining all the advantages, we get

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{pWPRF},R}(\lambda) &= |\Pr[\mathsf{G}_0] - \frac{1}{2}| \leq |\Pr[\mathsf{G}_0] - \Pr[\mathsf{G}_1]| + |\Pr[\mathsf{G}_1] - \frac{1}{2}| \\ &= \mathsf{Adv}_{\mathcal{D}}^{i\mathcal{O}}(\lambda) + \mathsf{Adv}_{\mathcal{B}}^{\mathsf{pPRF}}(\lambda) \\ &< \mathsf{negl}(\lambda) \quad \text{(by the assumptions in the theorem)}\end{aligned}$$

This completes the proof.

B Security of pWEPRF

Theorem 4 *The* pWEPRF = (Gen, F, PuncKey, PuncF, Eval) *described in Figure 5 with $\mathcal{O} = e\mathcal{O}$ is a selectively secure puncturable witness-extractable pseudorandom function if* pPRF *is a secure puncturable pseudorandom function and $e\mathcal{O}$ is an extractability obfuscator for the circuit class $\mathcal{C}_\lambda$.*

Proof. We prove the security by showing indistinguishability of the following games. We start with Game 0 which is the standard selective security experiment as in Def. 8. Let $\mathsf{G_i}$ be the event $b = b'$ in each Game i.
<u>Game 0 ⇒ Game 1</u>: Game 1 is exactly same as the Game 0 except we replace the circuit $C[\mathsf{K}]$ with a new circuit $C[\mathsf{K}, x^*]$ defined in Fig. 11. Suppose, the adversary $\mathcal{A}$ can distinguish between Game 0 and Game 1 with non-negligible advantage then

$$\mathsf{Adv}_{\mathcal{A}}^{\text{Game 0-1}}(\lambda) = |\ \Pr[\mathsf{G}_0] - \Pr[\mathsf{G}_1]| > \alpha(\lambda)$$

$x^* \leftarrow \mathcal{A}(1^\lambda)$
$\mathsf{K} \leftarrow \mathsf{pPRF.Gen}(1^\lambda)$
$\boxed{\widetilde{C} \leftarrow e\mathcal{O}(1^\lambda, C[\mathsf{K}, x^*])}$
set $\mathsf{ek} = \widetilde{C}$
$\mathsf{fk}_{x^*} \leftarrow \mathsf{pPRF.PuncKey}(\mathsf{K}, x^*)$
$y_0 \leftarrow \mathsf{pPRF.Eval}(\mathsf{K}, x^*), y_1 \leftarrow \mathcal{Y}$
$b \leftarrow \{0, 1\}$
$b' \leftarrow \mathcal{A}(\mathsf{ek}, \mathsf{fk}_{x^*}, y_b)$
return 1 if $b = b'$

<u>$C[\mathsf{K}, x^*](x, w)$</u>

if $R(x, w) = 1$
  if $x = x^*$
    return $\perp$
  else
    $y \leftarrow \mathsf{pPRF.Eval}(\mathsf{K}, x)$
    return y
else
  return $\perp$

Fig. 11: Game 1

for some non-negligible function α. We show below that there exists an extractor $\mathcal{E}$ and a polynomial $\mathsf{p}_{\mathcal{E}}$ such that $\mathcal{E}(1^\lambda, x^*)$ outputs a witness w^* satisfying

$R(x^*, w^*) = 1$ with overwhelming probability and runs in time $\mathsf{p}_{\mathcal{E}}(\lambda, 1/\beta)$ where $\beta = (\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game\ 0\text{-}1}}(\lambda) - \alpha(\lambda))$.

The two games differ only in the obfuscated circuits. So, we consider a PPT distinguisher $\mathcal{D}$ of $e\mathcal{O}$ as defined in Def. 13. Specifically, $\mathcal{D}$ collects two circuits from a circuit sampler $\mathsf{S}(1^\lambda, \cdot)$ and an obfuscated circuit (from it's challenger), then it simulates the security game for $\mathcal{A}$ as follows:

$\underline{\mathcal{D}(1^\lambda, \widetilde{C}, C[\mathsf{K}], C[\mathsf{K}, x^*], \mathsf{aux})}$:

1. parse $\mathsf{aux} = (\mathsf{fk}_{x^*}, y^*)$
2. set $\mathsf{ek} = \widetilde{C}, y_0 = y^*$
3. $y_1 \leftarrow \mathcal{Y}$
4. $b \leftarrow \{0, 1\}$
5. $b' \leftarrow \mathcal{A}(\mathsf{ek}, \mathsf{fk}_{x^*}, y_b)$
6. return 1 if $b = b'$

$\underline{\mathsf{S}(1^\lambda, x^*)}$

1. $\mathsf{K} \leftarrow \mathsf{pPRF.Gen}(1^\lambda)$
2. construct $C[\mathsf{K}], C[\mathsf{K}, x^*]$
3. $y^* \leftarrow \mathsf{pPRF.Eval}(\mathsf{K}, x^*)$
4. $\mathsf{fk}_{x^*} \leftarrow \mathsf{pPRF.PuncKey}(\mathsf{K}, x^*)$
5. set $\mathsf{aux} = (\mathsf{fk}_{x^*}, y^*)$
6. return $(C[\mathsf{K}], C[\mathsf{K}, x^*], \mathsf{aux})$

If $\widetilde{C} \leftarrow e\mathcal{O}(1^\lambda, C[\mathsf{K}])$, then $\mathcal{D}$ simulates $\mathsf{Game\ 0}$ and if $\widetilde{C} \leftarrow e\mathcal{O}(1^\lambda, C[\mathsf{K}, x^*])$, then $\mathcal{D}$ simulates $\mathsf{Game\ 1}$. Therefore, $\mathcal{D}$ can distinguish between the obfuscated circuits with the same advantage $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game\ 0\text{-}1}}(\lambda)$ of $\mathcal{A}$. By the extractability property of $e\mathcal{O}$, there exists an extractor $\mathcal{E}'$ and a polynomial $\mathsf{p}_{\mathcal{E}'}$ such that $\mathcal{E}'(1^\lambda, C[\mathsf{K}], C[\mathsf{K}, x^*], \mathsf{aux})$ outputs an input $(\bar{x}, \bar{w})$ at which the two circuits differ and runs in time $\mathsf{p}_{\mathcal{E}'}(\lambda, 1/\beta)$. Note that, the two circuits differs only when $\bar{x} = x^*$ and $R(x^*, \bar{w}) = 1$.

Thus, the extractor $\mathcal{E}(1^\lambda, x^*)$ of pWEPRF simply runs $\mathsf{S}(1^\lambda, x^*)$ to obtain $(C[\mathsf{K}], C[\mathsf{K}, x^*], \mathsf{aux})$ and then executes $\mathcal{E}'(1^\lambda, C[\mathsf{K}], C[\mathsf{K}, x^*], \mathsf{aux})$ to get a witness w^* such that $R(x^*, w^*) = 1$ holds with high probability. The runtime of $\mathcal{E}$ is equal to the runtime of S plus the runtime of $\mathcal{E}'$ and hence is bounded by $\mathrm{poly}(\lambda) + \mathsf{p}_{\mathcal{E}'}(\lambda, 1/\beta) = \mathsf{p}_{\mathcal{E}}(\lambda, 1/\beta)$ for some polynomial $\mathsf{p}_{\mathcal{E}}$ where $\beta = (\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game\ 0\text{-}1}}(\lambda) - \alpha(\lambda))$.

$\underline{\mathsf{Game\ 1} \Rightarrow \mathsf{Game\ 2}}$: In $\mathsf{Game\ 2}$, we set $\mathsf{ek} \leftarrow e\mathcal{O}(1^\lambda, C[\mathsf{fk}_{x^*}, x^*])$. The circuit $C[\mathsf{fk}_{x^*}, x^*]$ is the same as the circuit $C[\mathsf{K}, x^*]$ defined in Fig. 11 except that we compute $y \leftarrow \mathsf{pPRF.PuncEval}(\mathsf{fk}_{x^*}, x)$ in line 5. We see that the two circuits are functionally equivalent. Suppose, $(\bar{x}, \bar{w})$ be any arbitrary input to the circuits. If $\bar{x} \neq x^*$, then the circuits return the same value as $\mathsf{pPRF.Eval}(\mathsf{K}, \bar{x}) = \mathsf{pPRF.PuncEval}(\mathsf{fk}_{x^*}, \bar{x})$. If $\bar{x} = x^*$ then the circuits return $\perp$ because of the check in line 1 or 2. Therefore, by the extractability property of $e\mathcal{O}$ (Def. 13), we have

$$|\Pr[\mathsf{G}_1] - \Pr[\mathsf{G}_2]| = \mathsf{Adv}_{\mathcal{D}}^{e\mathcal{O}}(\lambda) = \mu(\lambda)$$

where μ is a negligible function of λ. If the advantage is not bounded by a negligible function of λ, then there exists an extractor $\mathcal{E}'$ which would produce an input where the two circuits differ, leading towards a contradiction as the circuits are equivalent.

Suppose, the advantage of $\mathcal{A}$ in $\mathsf{Game\ 2}$ is non-negligible. Then we construct an adversary $\mathcal{B}$ which will break the security of pPRF with the same advantage.

$\underline{\mathcal{B}(1^\lambda, x^*)}$:

1. send x^* to its challenger
2. The pPRF-challenger does the following:

(a) generate $\mathsf{K} \leftarrow \mathsf{pPRF.Gen}(1^\lambda)$
(b) compute $\mathsf{fk}_{x^*} \leftarrow \mathsf{pPRF.PuncKey}(\mathsf{K}, x^*)$
(c) set $y_0 \leftarrow \mathsf{pPRF.Eval}(\mathsf{K}, x^*)$ and $y_1 \leftarrow \mathcal{Y}$
(d) pick $b \leftarrow \{0,1\}$
(e) return (fk_{x^*}, y_b) to $\mathcal{B}$

3. compute $\widetilde{C} \leftarrow e\mathcal{O}(1^\lambda, C[\mathsf{fk}_{x^*}, x^*])$ and set $\mathsf{ek} = \widetilde{C}$
4. get $b' \leftarrow \mathcal{A}(\mathsf{ek}, \mathsf{fk}_{x^*}, y_b)$
5. return 1 if $b = b'$

Note that $\mathcal{B}$ perfectly simulates Game 2 for $\mathcal{A}$. If $\mathcal{A}$ can guess the bit b in Game 2 with a non-negligible advantage, then $\mathcal{B}$ breaks the security of pPRF with the same advantage. Therefore, the security of pPRF guarantees that

$$|\Pr[\mathsf{G}_2] - \tfrac{1}{2}| = \mathsf{Adv}_{\mathcal{B}}^{\mathsf{pPRF}}(\lambda)$$

Combining all the advantages we have

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{pWEPRF},R}(\lambda) &= |\Pr[\mathsf{G}_0] - \frac{1}{2}| \leq \sum_{i=0}^{1} |\Pr[\mathsf{G}_i] - \Pr[\mathsf{G}_{i+1}]| + |\Pr[\mathsf{G}_2] - \frac{1}{2}| \\ &= \mathsf{Adv}_{\mathcal{A}}^{\textsf{Game 0-1}}(\lambda) + \mu(\lambda) + \mathsf{Adv}_{\mathcal{B}}^{\mathsf{pPRF}}(\lambda) \\ &< \mathsf{Adv}_{\mathcal{A}}^{\textsf{Game 0-1}}(\lambda) + \mathsf{negl}(\lambda) \quad \text{(by the assumptions in the theorem)}\end{aligned}$$

Thus, $|\mathsf{Adv}_{\mathcal{A}}^{\mathsf{pWEPRF},R}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{\textsf{Game 0-1}}(\lambda)| < \mathsf{negl}(\lambda)$ implies $\mathsf{Adv}_{\mathcal{A}}^{\textsf{Game 0-1}}(\lambda) = \mathsf{Adv}_{\mathcal{A}}^{\mathsf{pWEPRF},R}(\lambda)$ excluding the negligible term. Hence, by the similar arguments as in the transition from Game 0 to Game 1, we conclude that if $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{pWEPRF},R}(\lambda) > \varrho(\lambda)$ for some non-negligible function ϱ, then there exists an extractor $\mathcal{E}$ and a polynomial $\mathsf{p}_{\mathcal{E}}$ such that $\mathcal{E}(1^\lambda, x^*)$ outputs a witness w^* satisfying $R(x^*, w^*) = 1$ with high probability and runs in time $\mathsf{p}_{\mathcal{E}}(\lambda, 1/\beta)$ where $\beta = (\mathsf{Adv}_{\mathcal{A}}^{\mathsf{pWEPRF},R}(\lambda) - \varrho(\lambda))$. This completes the proof.

C Offline Functional Witness Encryption

Definition 14 An offline functional witness encryption (OFWE) scheme for an NP language L with a relation R and a class of functions $\{\mathcal{F}_\lambda\}$ is a tuple of PPT algorithms (Setup, Enc, Dec) defined as follows:

- $(\mathsf{pp_e}, \mathsf{pp_d}) \leftarrow \mathsf{Setup}(1^\lambda, R)$: It is a randomized algorithm that takes as input a security parameter λ and a relation $R : \mathcal{X} \times \mathcal{W} \to \{0,1\}$, and produces two public parameters $\mathsf{pp_e}$ for encryption and $\mathsf{pp_d}$ for decryption.
- $c \leftarrow \mathsf{Enc}(\mathsf{pp_e}, x, f, m)$: It is a randomized algorithm that takes input a public parameter for encryption $\mathsf{pp_e}$, an element $x \in \mathcal{X}$, a function $f : \mathcal{M} \times \mathcal{W} \to \mathcal{M}'$ that belongs to $\mathcal{F}_\lambda$ and a message $m \in \mathcal{M}$, and outputs a ciphertext c.
- $\mathsf{Dec}(\mathsf{pp_d}, c, w) \in \mathcal{M}' \cup \{\perp\}$: It is a deterministic algorithm that takes as input a public parameter for decryption $\mathsf{pp_d}$, a ciphertext c and a witness $w \in \mathcal{W}$, and outputs either an element $m' \in \mathcal{M}'$ or $\perp$.

1. $x^* \leftarrow \mathcal{A}(1^\lambda)$
2. $(\mathsf{pp_e}, \mathsf{pp_d}) \leftarrow \mathsf{Setup}(1^\lambda, R)$
3. $((f_0, m_0), (f_1, m_1)) \leftarrow \mathcal{A}(\mathsf{pp_e}, \mathsf{pp_d})$
4. $b \leftarrow \{0, 1\}$
5. $c \leftarrow \mathsf{Enc}(\mathsf{pp_e}, x^*, f_b, m_b)$
6. $b' \leftarrow \mathcal{A}(c)$
7. return 1 if $(b' = b) \wedge (|f_0| + |m_0| = |f_1| + |m_1|) \wedge (f_0(m_0, w) = f_1(m_1, w) \forall w \in \mathcal{W}$ s.t. $R(x^*, w) = 1)$

Fig. 12: $\mathsf{Expt}_{\mathcal{A}}^{\mathsf{OFWE},R}(1^\lambda, b)$

1. $(x^*, f, m_0, m_1) \leftarrow \mathcal{A}(1^\lambda)$
2. $(\mathsf{pp_e}, \mathsf{pp_d}) \leftarrow \mathsf{Setup}(1^\lambda, R)$
3. $b \leftarrow \{0, 1\}$
4. $c \leftarrow \mathsf{Enc}(\mathsf{pp_e}, x^*, f, m_b)$
5. $b' \leftarrow \mathcal{A}(\mathsf{pp_e}, \mathsf{pp_d}, c)$
6. return 1 if $(b' = b) \wedge (|m_0| = |m_1|)$

Fig. 13: $\mathsf{Expt}_{\mathcal{A}}^{\mathsf{EOFWE},R}(1^\lambda, b)$

The OFWE scheme is said to be correct if the following holds:

– *correctness*: For all $\lambda \in \mathbb{N}, x \in \mathcal{X}$, $w \in \mathcal{W}$, $m \in \mathcal{M}, f \in \mathcal{F}_\lambda$ and $(\mathsf{pp_e}, \mathsf{pp_d}) \leftarrow \mathsf{Setup}(1^\lambda, R)$, we require that

$$\Pr[\mathsf{Dec}(\mathsf{pp_d}, \mathsf{Enc}(\mathsf{pp_e}, x, f, m), w) = f(m, w) : R(x, w) = 1] = 1$$

We consider semi-adaptive security model for OFWE described in the experiment $\mathsf{Expt}_{\mathcal{A}}^{\mathsf{OFWE},R}(1^\lambda)$ (Fig. 12).

Definition 15 An offline functional witness encryption OFWE for an NP language L with a relation R and a class of functions $\{\mathcal{F}_\lambda\}$ is said to be semi-adaptively secure if, for all PPT adversary $\mathcal{A}$, there exists a negligible function negl such that

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{OFWE},R}(\lambda) = |\Pr[\mathsf{Expt}_{\mathcal{A}}^{\mathsf{OFWE},R}(1^\lambda) = 1] - \tfrac{1}{2}| < \mathsf{negl}(\lambda)$$

For extractable offline functional witness encryption we consider security in selective model where $\mathcal{A}$ has to submit a challenge tuple (x^*, f, m_0, m_1) before the setup. We call this experiment as $\mathsf{Expt}_{\mathcal{A}}^{\mathsf{EOFWE},R}(1^\lambda)$ defined in Fig. 13.

Definition 16 An offline functional witness encryption OFWE is said to be selectively secure extractable offline functional witness encryption (EOFWE) for an NP language L with a relation R and a class of functions $\{\mathcal{F}_\lambda\}$, if for any PPT adversary $\mathcal{A}$ there exists an extractor $\mathcal{E}$ and a polynomial $\mathsf{p}_{\mathcal{E}}$ such that, if

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{EOFWE},R}(\lambda) = |\Pr[\mathsf{Expt}_{\mathcal{A}}^{\mathsf{EOFWE},R}(1^\lambda) = 1] - \tfrac{1}{2}| > \alpha(\lambda)$$

for some non-negligible function α, then $\mathcal{E}(1^\lambda, (x^*, f, m_0, m_1))$ outputs a witness $w^* \in \mathcal{W}$ such that $R(x^*, w^*) = 1$ and $f(m_0, w^*) \neq f(m_1, w^*)$ hold with overwhelming probability and $\mathcal{E}$ runs in time $\mathsf{p}_{\mathcal{E}}(\lambda, 1/\beta)$ where (x^*, f, m_0, m_1) is the challenge tuple and $\beta = (\mathsf{Adv}_{\mathcal{A}}^{\mathsf{EOFWE},R}(\lambda) - \alpha(\lambda))$.

C.1 Construction: (Extractable) Offline Functional Witness Encryption

Here, we present our construction of OFWE = (Setup, Enc, Dec) for an NP language L with a relation $R : \mathcal{X} \times \mathcal{W} \rightarrow \{0, 1\}$ and a class of functions $\{\mathcal{F}_\lambda\}$.

Setup$(1^\lambda, R)$:
1. $(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{pWPRF.Gen}(1^\lambda, R')$
2. $\widetilde{C} \leftarrow \mathcal{O}(1^\lambda, C[\mathsf{fk}])$
3. set $\mathsf{pp_e} = \mathsf{ek}$, $\mathsf{pp_d} = \widetilde{C}$
4. return $(\mathsf{pp_e}, \mathsf{pp_d})$

Enc$(\mathsf{pp_e}, x, f, m)$:
1. parse $\mathsf{pp_e} = \mathsf{ek}$
2. $u \leftarrow \{0,1\}^\lambda$, $v \leftarrow \mathsf{PRG}(x \oplus u)$
3. $y \leftarrow \mathsf{pWPRF.Eval}(\mathsf{ek}, (x, v), u)$
4. $\mathsf{K} \leftarrow \mathsf{SKE.Gen}(1^\lambda; y)$
5. $c_s \leftarrow \mathsf{SKE.Enc}(\mathsf{K}, (f, m))$
6. return $c = (c_s, x, v)$

$C[\mathsf{fk}](c, w)$
1. parse $c = (c_s, x, v)$
2. if $R(x, w) = 1$
3. $\quad y \leftarrow \mathsf{pWPRF.F}(\mathsf{fk}, (x, v))$
4. $\quad \mathsf{K} \leftarrow \mathsf{SKE.Gen}(1^\lambda; y)$
5. $\quad (f, m) \leftarrow \mathsf{SKE.Dec}(\mathsf{K}, c_s)$
6. $\quad$ return $f(m, w)$
7. else
8. $\quad$ return $\perp$

Dec$(\mathsf{pp_d}, c, w)$:
1. parse $\mathsf{pp_d} = \widetilde{C}$
2. return $\widetilde{C}(c, w)$

Fig. 14: Construction of OFWEs with optimal ciphertexts where $\mathcal{O}$ is either $i\mathcal{O}$ for normal OFWE or $e\mathcal{O}$ for extractable OFWE (EOFWE)

We consider the statement space $\mathcal{X}$ to be $\{0,1\}^\lambda$ and $\mathcal{W} = \{0,1\}^n$ where n is a polynomial in the security parameter λ. We utilize the following set of primitives for our construction:

- A pseudorandom generator $\mathsf{PRG} : \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$.
- A CIND secure symmetric key encryption SKE = (Gen, Enc, Dec).
- A pWPRF = (Gen, F, PuncKey, PuncF, Eval) for the NP language $L' = \{(x, v) : \exists u \in \{0,1\}^\lambda \text{ such that } \mathsf{PRG}(x \oplus u) = v\}$ with a relation $R' : \mathcal{X}' \times \mathcal{W}' \to \{0,1\}$. So, $R'((x, v), u) = 1$ if $\mathsf{PRG}(x \oplus u) = v$, 0 otherwise.
- An obfuscator $\mathcal{O}$ for the class of circuits $\mathcal{C}_\lambda$ required in the constructions. The only difference between the constructions of OFWE and extractable OFWE (EOFWE) is that: $\mathcal{O}$ is an indistinguishability obfuscator ($i\mathcal{O}$) for OFWE whereas $\mathcal{O}$ is an extractability obfuscator ($e\mathcal{O}$) for EOFWE.

Our OFWE construction is described in Fig. 3. We assume that the circuit $C[\mathsf{fk}] \in \mathcal{C}_\lambda$ and $\mathcal{O}$ is an $i\mathcal{O}$. For correctness, we need to ensure that the same key $\mathsf{K} \leftarrow \mathsf{SKE.Gen}(1^\lambda; y)$ is generated during encryption and decryption of OFWE. Note that, we evaluate y using the $\mathsf{pWPRF.Eval}(\mathsf{ek}, \cdot, \cdot)$ with a statement (x, v) and a witness u such that $R'((x, v), u) = 1$. In decryption, we generate y inside the circuit $\widetilde{C}$ using $\mathsf{pWPRF.F}(\mathsf{fk}, \cdot)$ with the statement (x, v) extracted from the ciphertext. By the correctness of Eval, we ensure that the same randomness y is used while decryption and hence $\mathsf{SKE.Dec}(\mathsf{K}, c_s)$ returns (f, m) that was encrypted in Enc if $R(x, w) = 1$. Finally, the functionality of $i\mathcal{O}$ guarantees that $\widetilde{C}$ returns $f(m, w)$ as required.
Efficiency: The ciphertext size of our OFWEs is also compact. Excluding the size of the instance, the ciphertext size can be written as $|c_s| + |v| = |m| + |f| + 2\lambda$ where $|m|, |f|$ denote the size of message and function respectively. Note that, in SKE the size of ciphertexts are usually equal to the size of plaintexts. Hence,

$x^* \leftarrow \mathcal{A}(1^\lambda)$
$(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{pWPRF.Gen}(1^\lambda, R')$
$\widetilde{C} \leftarrow i\mathcal{O}(1^\lambda, C[\mathsf{fk}])$
set $\mathsf{pp_e} = \mathsf{ek}, \mathsf{pp_d} = \widetilde{C}$
$((f_0, m_0), (f_1, m_1)) \leftarrow \mathcal{A}(\mathsf{pp_e}, \mathsf{pp_d})$
$u \leftarrow \{0,1\}^\lambda, v \leftarrow \mathsf{PRG}(x^* \oplus u)$
$\boxed{y^* \leftarrow \mathsf{pWPRF.F}(\mathsf{fk}, (x^*, v))}$
$\mathsf{K}^* \leftarrow \mathsf{SKE.Gen}(1^\lambda; y^*)$
$b \leftarrow \{0,1\}$
$c_s \leftarrow \mathsf{SKE.Enc}(\mathsf{K}^*, (f_b, m_b))$
set $c = (c_s, x^*, v)$
$b' \leftarrow \mathcal{A}(c)$
return 1 if $(b = b')$

Fig. 15: Game 1

$x^* \leftarrow \mathcal{A}(1^\lambda)$
$(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{pWPRF.Gen}(1^\lambda, R')$
$\widetilde{C} \leftarrow i\mathcal{O}(1^\lambda, C[\mathsf{fk}])$
set $\mathsf{pp_e} = \mathsf{ek}, \mathsf{pp_d} = \widetilde{C}$
$((f_0, m_0), (f_1, m_1)) \leftarrow \mathcal{A}(\mathsf{pp_e}, \mathsf{pp_d})$
$\boxed{v \leftarrow \{0,1\}^{2\lambda}}$
$y^* \leftarrow \mathsf{pWPRF.F}(\mathsf{fk}, (x^*, v))$
$\mathsf{K}^* \leftarrow \mathsf{SKE.Gen}(1^\lambda; y^*)$
$b \leftarrow \{0,1\}$
$c_s \leftarrow \mathsf{SKE.Enc}(\mathsf{K}^*, (f_b, m_b))$
set $c = (c_s, x^*, v)$
$b' \leftarrow \mathcal{A}(c)$
return 1 if $(b = b')$

Fig. 16: Game 2

the ciphertext size of OFWE is *optimal*. To encrypt a larger message with an arbitrary function, one can split the plaintext into blocks of equal length (as supported by the SKE) and then use a suitable modes of operation to encrypt it with the same key K. We use the same key K to decrypt the ciphertext of SKE and get back the original message. The size of the public parameter for encryption ek (or $\mathsf{pp_e}$) is independent of the prime relation R. It depends on the fixed relation R' which verifies only a PRG computation. Hence, our OFWE encryption is the most efficient among the existing constructions.

Theorem 5 *The* OFWE = (Setup, Enc, Dec) *described in Figure 14 with* $\mathcal{O} = i\mathcal{O}$ *is a semi-adaptively secure offline functional witness encryption if* PRG *is a secure pseudorandom generator,* pWPRF *is a selectively secure puncturable witness pseudorandom function,* $i\mathcal{O}$ *is an indistinguishability obfuscator for the circuit class* $\mathcal{C}_\lambda$ *and* SKE *is a* CIND *secure symmetric key encryption.*

Proof. The proof is partly similar to the proof of Th. 1. In contrast to a normal OWE, here we are allowing decryption for the challenge statement x^* whenever $f_0(m_0, w) = f_1(m_1, w)$ holds for a witness w satisfying $R(x^*, w) = 1$.

We start with Game 0 which is the standard security experiment $\mathsf{Expt}_{\mathcal{A}}^{\mathsf{OFWE},R}(1^\lambda)$ as defined in Fig. 12. For Game i, we denote by $\mathsf{G_i}$ the event $b = b'$. In each game, we assume $\mathcal{A}$ submits $(f_0, m_0), (f_1, m_1) \in \mathcal{F}_\lambda \times \mathcal{M}$ such that $|f_0| + |m_0| = |f_1| + |m_1|$ and for all $w \in \mathcal{W}$ satisfying $R(x^*, w) = 1$ it holds that $f_0(m_0, w) = f_1(m_1, w)$. The circuits used in the security proof are assumed to be padded to a fixed maximum size.

Game 0 $\Rightarrow$ Game 1: In Game 0, we generate the encryption key as $\mathsf{K} \leftarrow \mathsf{SKE.Gen}$ $(1^\lambda; y)$ where $y^* \leftarrow \mathsf{pWPRF.Eval}(\mathsf{ek}, (x^*, v), u)$. But, Game 1 (Fig. 15) directly sets $y^* \leftarrow \mathsf{pWPRF.F}(\mathsf{fk}, (x^*, v))$ without using the witness u. By the correctness Eval:

$$\mathsf{pWPRF.Eval}(\mathsf{ek}, (x^*, v), u) = \mathsf{pWPRF.F}(\mathsf{fk}, (x^*, v)) \text{ as } R'((x^*, v), u) = 1.$$

It is clear that the distribution of ciphertexts in both the games are identical and hence we have $\Pr[\mathsf{G_0}] = \Pr[\mathsf{G_1}]$.

Game 1 $\Rightarrow$ Game 2: In Game 2, described in Fig. 16, we pick v uniformly at random from $\{0,1\}^{2\lambda}$ instead of computing $v \leftarrow \mathsf{PRG}(x^* \oplus u)$. Note that, given x^*, the distribution of $x^* \oplus u$ is uniform over $\{0,1\}^\lambda$ for $u \leftarrow \{0,1\}^\lambda$. By the security of PRG (Def. 1), the distinguishing advantage of $\mathcal{A}$ between Game 1 and Game 2 is written as

$$|\Pr[\mathsf{G}_1] - \Pr[\mathsf{G}_2]| = \mathsf{Adv}^{\mathsf{PRG}}_{\mathcal{B}_1}(\lambda)$$

where $\mathcal{B}_1$ is a PRG-adversary.

Game 2 $\Rightarrow$ Game 3: We describe Game 3 in Fig. 17 where we replace the circuit $C[\mathsf{fk}]$ by the circuit $C[\mathsf{fk}_{z^*}, \mathsf{K}^*, z^*]$ and set the public parameter for decryption as $\mathsf{pp_d} \leftarrow i\mathcal{O}(1^\lambda, C[\mathsf{fk}_{z^*}, \mathsf{K}^*, z^*])$. The new circuit $C[\mathsf{fk}_{z^*}, \mathsf{K}^*, z^*]$ works as follows:

$C[\mathsf{fk}_{z^*}, \mathsf{K}^*, z^*](c, w)$

parse $c = (c_s, x, v)$
if $R(x, w) = 1$
    if $(x, v) = z^*$
      $(f, m) \leftarrow \mathsf{SKE.Dec}(\mathsf{K}^*, c_s)$
      return $f(m, w)$
    else $y \leftarrow \mathsf{pWPRF.PuncF}(\mathsf{fk}_{z^*}, (x, v))$
          $\mathsf{K} \leftarrow \mathsf{SKE.Gen}(1^\lambda; y)$
          $(f, m) \leftarrow \mathsf{SKE.Dec}(\mathsf{K}^*, c_s)$
          return $f(m, w)$
else
    return $\perp$

Note that, the two circuits $C[\mathsf{fk}]$ and $C[\mathsf{fk}_{z^*}, \mathsf{K}^*, z^*]$ are functionally equivalent. Let $(\bar{c}, \bar{w})$ be any arbitrary input where $\bar{c} = (\bar{c}_s, \bar{x}, \bar{v})$. If $(\bar{x}, \bar{v}) = z^*$, then both the circuits use the K^* to decrypt $\bar{c}_s$ whenever $R(x^*, \bar{w}) = 1$ holds; otherwise output $\perp$. If $(\bar{x}, \bar{v}) \neq z^*$, then by the correctness of PuncF we have

$$\mathsf{pWPRF.F}(\mathsf{fk}, (\bar{x}, \bar{v})) = \mathsf{pWPRF.PuncF}(\mathsf{fk}_{z^*}, (\bar{x}, \bar{v}))$$

and hence $C[\mathsf{fk}](\bar{c}, \bar{w}) = C[\mathsf{fk}_{z^*}, \mathsf{K}^*, z^*](\bar{c}, \bar{w})$. Therefore, by the indistinguishability property of $i\mathcal{O}$, we have

$$|\Pr[\mathsf{G}_2] - \Pr[\mathsf{G}_3]| = \mathsf{Adv}^{i\mathcal{O}}_{\mathcal{D}}(\lambda)$$

where $\mathcal{D}$ is a PPT distinguisher for $i\mathcal{O}$.

Game 3 $\Rightarrow$ Game 4: In Game 4, described in Fig. 18, we sample y uniformly at random from $\mathcal{Y}$ which is the co-domain of $\mathsf{pWPRF.F}(\mathsf{fk}, \cdot)$. We need to show that if $\mathcal{A}$ is able to distinguish between these two games, then there is an adversary $\mathcal{B}_2$ which will break the selective security of pWPRF (defined in Fig. 1) with the same advantage. Let $z^* = (x^*, v)$ be the challenge statement of $\mathcal{B}_2$ for a random $v \leftarrow \{0,1\}^{2\lambda}$.

$\mathcal{B}_2(1^\lambda, z^*)$:

1. send z^* to its challenger
2. The pWPRF-challenger does the following:
   (a) generate $(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{pWPRF.Gen}(1^\lambda, R')$
   (b) compute a punctured key $\mathsf{fk}_{z^*} \leftarrow \mathsf{pWPRF.PuncKey}(\mathsf{fk}, z^*)$

1. $x^* \leftarrow \mathcal{A}(1^\lambda)$
2. $(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{pWPRF.Gen}(1^\lambda, R')$
3. $v \leftarrow \{0,1\}^{2\lambda}$, set $z^* = (x^*, v)$
4. $\mathsf{fk}_{z^*} \leftarrow \mathsf{pWPRF.PuncKey}(\mathsf{fk}, z^*)$
5. $y^* \leftarrow \mathsf{pWPRF.F}(\mathsf{fk}, (x^*, v))$
6. $\mathsf{K}^* \leftarrow \mathsf{SKE.Gen}(1^\lambda; y)$
7. $\boxed{\widetilde{C} \leftarrow i\mathcal{O}(1^\lambda, C[\mathsf{fk}_{z^*}, \mathsf{K}^*, z^*])}$
8. set $\mathsf{pp_e} = \mathsf{ek}, \mathsf{pp_d} = \widetilde{C}$
9. $((f_0, m_0), (f_1, m_1)) \leftarrow \mathcal{A}(\mathsf{pp_e}, \mathsf{pp_d})$
10. $b \leftarrow \{0,1\}$
11. $c_s \leftarrow \mathsf{SKE.Enc}(\mathsf{K}^*, (f_b, m_b))$
12. set $c = (c_s, x^*, v)$
13. $b' \leftarrow \mathcal{A}(c)$
14. return 1 if $(b = b')$

Fig. 17: Game 3

1. $x^* \leftarrow \mathcal{A}(1^\lambda)$
2. $(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{pWPRF.Gen}(1^\lambda, R')$
3. $v \leftarrow \{0,1\}^{2\lambda}$, set $z^* = (x^*, v)$
4. $\mathsf{fk}_{z^*} \leftarrow \mathsf{pWPRF.PuncKey}(\mathsf{fk}, z^*)$
5. $\boxed{y^* \leftarrow \mathcal{Y}}$
6. $\mathsf{K}^* \leftarrow \mathsf{SKE.Gen}(1^\lambda; y)$
7. $\widetilde{C} \leftarrow i\mathcal{O}(1^\lambda, C[\mathsf{fk}_{z^*}, \mathsf{K}^*, z^*])$
8. set $\mathsf{pp_e} = \mathsf{ek}, \mathsf{pp_d} = \widetilde{C}$
9. $((f_0, m_0), (f_1, m_1)) \leftarrow \mathcal{A}(\mathsf{pp_e}, \mathsf{pp_d})$
10. $b \leftarrow \{0,1\}$
11. $c_s \leftarrow \mathsf{SKE.Enc}(\mathsf{K}^*, (f_b, m_b))$
12. set $c = (c_s, x^*, v)$
13. $b' \leftarrow \mathcal{A}(c)$
14. return 1 if $(b = b')$

Fig. 18: Game 4

(c) set $y_0 \leftarrow \mathsf{pWPRF.F}(\mathsf{fk}, z^*)$ and $y_1 \leftarrow \mathcal{Y}$
(d) pick $\tilde{b} \leftarrow \{0,1\}$
(e) return $(\mathsf{ek}, \mathsf{fk}_{z^*}, y_{\tilde{b}})$ to $\mathcal{B}_2$

3. compute the encryption key as $\mathsf{K}^* \leftarrow \mathsf{SKE.Gen}(1^\lambda; y_{\tilde{b}})$
4. compute $\widetilde{C} \leftarrow i\mathcal{O}(1^\lambda, C[\mathsf{fk}_{z^*}, \mathsf{K}^*, z^*])$ and set $\mathsf{pp_e} = \mathsf{ek}, \mathsf{pp_d} = \widetilde{C}$
5. receive $((f_0, m_0), (f_1, m_1)) \leftarrow \mathcal{A}(\mathsf{pp_e}, \mathsf{pp_d})$
6. pick $b \leftarrow \{0,1\}$
7. compute the ciphertext as $c_s \leftarrow \mathsf{SKE.Enc}(\mathsf{K}^*, (f_b, m_b))$
8. set $c = (c_s, x^*, v)$
9. get $b' \leftarrow \mathcal{A}(c)$
10. return 1 if $(b = b')$

It is important to observe that $z^* = (x^*, v) \notin L'$ with overwhelming probability. Since $v \leftarrow \{0,1\}^{2\lambda}$, the probability that $\mathsf{PRG}(x^* \oplus u) = v$ for some u drawn uniformly at random from $\{0,1\}^\lambda$ is at most $2^{-\lambda}$ which is negligible in λ. So, $\mathcal{B}_2$ is an honest pWPRF-adversary.

If the pWPRF-challenger picks $\tilde{b} = 0$ then $\mathcal{B}_2$ simulates $\mathsf{Game\ 3}$, and if it chooses $\tilde{b} = 1$ then $\mathcal{B}_2$ simulates $\mathsf{Game\ 4}$. Therefore, the advantage of $\mathcal{A}$ in distinguishing between $\mathsf{Game\ 3}$ and $\mathsf{Game\ 4}$ is the same as the advantage of $\mathcal{B}_2$ in breaking the selective security of pWPRF. We get the following:

$$|\Pr[\mathsf{G}_3] - \Pr[\mathsf{G}_4]| = \mathsf{Adv}_{\mathcal{B}_2}^{\mathsf{pWPRF}, R'}(\lambda)$$

Finally, we note that in $\mathsf{Game\ 4}$, the encryption key is computed as $\mathsf{K}^* \leftarrow \mathsf{SKE.Gen}(1^\lambda; y^*)$ where y^* is sampled uniformly and independently from $\mathcal{Y}$. Therefore, by the CIND security of SKE (Def. 5) we have

$$|\Pr[\mathsf{G}_4] - \tfrac{1}{2}| = \mathsf{Adv}_{\mathcal{B}_3}^{\mathsf{SKE}}(\lambda)$$

where $\mathcal{B}_3$ is an adversary of CIND security game. Combining all the probabilities we have

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{OFWE},R}(\lambda) = |\Pr[\mathsf{G}_0] - \frac{1}{2}| &\leq \sum_{i=0}^{3} |\Pr[\mathsf{G}_i] - \Pr[\mathsf{G}_{i+1}]| + |\Pr[\mathsf{G}_4] - \frac{1}{2}| \\ &= \mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{PRG}}(\lambda) + \mathsf{Adv}_{\mathcal{D}}^{i\mathsf{O}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_2}^{\mathsf{pWPRF},R'}(\lambda) + \mathsf{Adv}_{\mathcal{B}_3}^{\mathsf{SKE}}(\lambda) \\ &< \mathsf{negl}(\lambda) \quad \text{(by the assumptions in the theorem)}\end{aligned}$$

This completes the proof.

Next, we discuss security of extractable OFWE in the following theorem.

Theorem 6 *The* $\mathsf{EOFWE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$ *described in Figure 14 with* $\mathcal{O} = e\mathcal{O}$ *is a selectively secure extractable offline functional witness encryption if* PRG *is a secure pseudorandom generator,* pWPRF *is a selectively secure puncturable witness pseudorandom function,* $e\mathcal{O}$ *is an extractability obfuscator for the circuit class* $\mathcal{C}_\lambda$ *and* SKE *is a* CIND *secure symmetric key encryption.*

Proof. We begin the proof with the standard EOFWE experiment $\mathsf{Expt}_{\mathcal{A}}^{\mathsf{EOFWE},R}(1^\lambda)$ which is described in Def. 16. Here, we name it as $\mathsf{EGame\ 0}$ and denote the security games by $\mathsf{EGame\ i}$. In each $\mathsf{EGame\ i}$, we consider $\mathsf{EG_i}$ as the event $b = b'$. We assume that $\mathcal{A}$ submits a challenge tuple (x^*, f, m_0, m_1) such that $|m_0| = |m_1|$ and the circuits used in the proof are padded to a maximum size.
$\underline{\mathsf{EGame\ 0} \Rightarrow \mathsf{EGame\ 1}}$: $\mathsf{EGame\ 1}$ is exactly the same as $\mathsf{EGame\ 0}$ except we replace the circuit $C[\mathsf{fk}]$ with a new circuit $C[\mathsf{fk}, X^*]$ defined in Fig. 19 where $X^* = (x^*, f, m_0, m_1)$. Suppose, the adversary $\mathcal{A}$ can distinguish between $\mathsf{EGame\ 0}$ and $\mathsf{EGame\ 1}$ with an advantage

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{EGame\ 0\text{-}1}}(\lambda) = |\ \Pr[\mathsf{EG}_0]\ -\ \Pr[\mathsf{EG}_1]| > \alpha(\lambda)$$

for some non-negligible function α. Then, we build an extractor $\mathcal{E}$ such that $\mathcal{E}(1^\lambda, X^*)$ outputs a witness w^* satisfying $R(x^*, w^*) = 1$ and $f(m_0, w^*) \neq f(m_1, w^*)$ with overwhelming probability and $\mathcal{E}$ runs in time $\mathsf{p}_{\mathcal{E}}(\lambda, 1/\beta)$ where $\mathsf{p}_{\mathcal{E}}$ a polynomial with $\beta = (\mathsf{Adv}_{\mathcal{A}}^{\mathsf{EGame\ 0\text{-}1}}(\lambda) - \alpha(\lambda))$.

We note that two games differ only in the obfuscated circuits. Thus, we consider a PPT distinguisher $\mathcal{D}$ of $e\mathcal{O}$ as defined in Def. 13. In particular, $\mathcal{D}$ collects two circuits from a circuit sampler $\mathsf{S}(1^\lambda, \cdot)$ and an obfuscated circuit (from it's challenger), then it simulates the security game for $\mathcal{A}$ as follows:

$\underline{\mathcal{D}(1^\lambda, \widetilde{C}, C[\mathsf{fk}], C[\mathsf{fk}, X^*], \mathsf{aux})}$:

1. parse $\mathsf{aux} = (\mathsf{ek}, X^*)$
2. parse $X^* = (x^*, f, m_0, m_1)$
3. set $\mathsf{pp_e} = \mathsf{ek}, \mathsf{pp_d} = \widetilde{C}$
4. follow steps 6-10 as in $\mathsf{EGame\ 1}$
5. set $c = (c_s, x^*, v)$
6. $b' \leftarrow \mathcal{A}(\mathsf{pp_e}, \mathsf{pp_d}, c)$
7. return 1 if $b = b'$

$\underline{\mathsf{S}(1^\lambda, X^*)}$

1. $(\mathsf{fk},\ \mathsf{ek}) \leftarrow \mathsf{pWPRF.Gen}(1^\lambda, R')$
2. construct $C[\mathsf{fk}], C[\mathsf{fk}, X^*]$
3. set $\mathsf{aux} = (\mathsf{ek}, X^*)$
4. return $(C[\mathsf{fk}], C[\mathsf{fk}, X^*], \mathsf{aux})$

1. $(x^*, f, m_0, m_1) \leftarrow \mathcal{A}(1^\lambda)$
2. $(\mathsf{fk}, \mathsf{ek}) \leftarrow \mathsf{pWPRF.Gen}(1^\lambda, R')$
3. set $X^* = (x^*, f, m_0, m_1)$
4. $\boxed{\widetilde{C} \leftarrow e\mathcal{O}(1^\lambda, C[\mathsf{fk}, X^*])}$
5. set $\mathsf{pp_e} = \mathsf{ek}, \mathsf{pp_d} = \widetilde{C}$
6. $u \leftarrow \{0,1\}^\lambda, v \leftarrow \mathsf{PRG}(x^* \oplus u)$
7. $y^* \leftarrow \mathsf{pWPRF.Eval}(\mathsf{ek}, (x^*, v), u)$
8. $\mathsf{K}^* \leftarrow \mathsf{SKE.Gen}(1^\lambda; y^*)$
9. $b \leftarrow \{0,1\}$
10. $c_s \leftarrow \mathsf{SKE.Enc}(\mathsf{K}^*, (f, m_b))$
11. set $c = (c_s, x^*, v)$
12. $b' \leftarrow \mathcal{A}(\mathsf{pp_e}, \mathsf{pp_d}, c)$
13. return 1 if $b = b'$

$\underline{C[\mathsf{fk}, X^*](c, w)}$

1. parse $c = (c_s, x, v)$
2. if $R(x, w) = 1$
3. $\quad y \leftarrow \mathsf{pWPRF.F}(\mathsf{fk}, (x, v))$
4. $\quad \mathsf{K} \leftarrow \mathsf{SKE.Gen}(1^\lambda; y)$
5. $\quad (\hat{f}, \hat{m}) \leftarrow \mathsf{SKE.Dec}(\mathsf{K}, c_s)$
6. $\quad$ if $(x = x^*) \wedge (f = \hat{f}) \wedge (f(m_0, w) \neq f(m_1, w))$
7. $\quad\quad$ return $\perp$
8. $\quad$ else
9. $\quad\quad$ return $\hat{f}(\hat{m}, w)$
10. else
11. $\quad$ return $\perp$

Fig. 19: EGame 1

If $\widetilde{C} \leftarrow e\mathcal{O}(1^\lambda, C[\mathsf{fk}])$ then $\mathcal{D}$ simulates EGame 0 and if $\widetilde{C} \leftarrow e\mathcal{O}(1^\lambda, C[\mathsf{fk}, X^*])$ then $\mathcal{D}$ simulates EGame 1. Therefore, $\mathcal{D}$ can distinguish between the obfuscated circuits with the same advantage of $\mathcal{A}$ in distinguishing EGame 0 and EGame 1. By the extractability property of $e\mathcal{O}$ (Def. 13), there exists an extractor $\mathcal{E}'$ and a polynomial $\mathsf{p}_{\mathcal{E}'}$ such that $\mathcal{E}'(1^\lambda, C[\mathsf{fk}], C[\mathsf{fk}, X^*], \mathsf{aux})$ outputs an input $(\bar{c}, \bar{w})$ at which the two circuits differ and runs in time $\mathsf{p}_{\mathcal{E}'}(\lambda, 1/\beta)$ with $\beta = (\mathsf{Adv}_{\mathcal{A}}^{\mathsf{EGame\ 0\text{-}1}}(\lambda) - \alpha(\lambda))$. Note that, the two circuits differ only when $\bar{c} = (\bar{c}_s, x^*, \bar{v})$ is well formed and $\bar{c}_s$ is an encryption of (f, m) such that $f(m_0, \bar{w}) \neq f(m_1, \bar{w})$ with $R(x^*, \bar{w}) = 1$.

Now, the extractor $\mathcal{E}(1^\lambda, X^*)$ of EOFWE simply runs $\mathsf{S}(1^\lambda, X^*)$ to obtain $(C[\mathsf{fk}], C[\mathsf{fk}, X^*], \mathsf{aux})$ and then executes $\mathcal{E}'(1^\lambda, C[\mathsf{fk}], C[\mathsf{fk}, X^*], \mathsf{aux})$ to get a witness w^* satisfying $R(x^*, w^*) = 1$ and $f(m_0, w^*) \neq f(m_1, w^*)$ with high probability. The runtime of $\mathcal{E}$ is equal to the runtime of S plus the runtime of $\mathcal{E}'$, hence is bounded by $\mathrm{poly}(\lambda) + \mathsf{p}_{\mathcal{E}'}(\lambda, 1/\beta) = \mathsf{p}_{\mathcal{E}}(\lambda, 1/\beta)$ for some polynomial $\mathsf{p}_{\mathcal{E}}$ where $\beta = (\mathsf{Adv}_{\mathcal{A}}^{\mathsf{EGame\ 0\text{-}1}}(\lambda) - \alpha(\lambda))$.

<u>EGame 1 $\Rightarrow$ EGame 2</u>: EGame 2 is exactly the same as EGame 1 except in line 7 of Fig. 19 where we compute $y \leftarrow \mathsf{pWPRF.F}(\mathsf{fk}, (x^*, v))$. By the correctness of Eval (using the same argument as in the transition from Game 0 to Game 1 of Th. 5), we have $\Pr[\mathsf{EG}_1] = \Pr[\mathsf{EG}_2]$.

<u>EGame 2 $\Rightarrow$ EGame 3</u>: In EGame 3, we choose $v \leftarrow \{0,1\}^{2\lambda}$ instead of computing $v \leftarrow \mathsf{PRG}(x^* \oplus u)$ as in EGame 2. The distribution of $x^* \oplus u$ is uniform over $\{0,1\}^\lambda$ as u is sampled uniformly at random from $\{0,1\}^\lambda$. Hence, the security of PRG (Def. 1) implies that

$$|\Pr[\mathsf{EG}_2] - \Pr[\mathsf{EG}_3]| = \mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{PRG}}(\lambda)$$

where $\mathcal{B}_1$ is a PRG-adversary.

<u>EGame 3 $\Rightarrow$ EGame 4</u>: In EGame 4, we set $\mathsf{pp_d} \leftarrow e\mathcal{O}(1^\lambda, C[\mathsf{fk}_{z^*}, \mathsf{K}^*, X^*])$ where $\mathsf{fk}_{z^*} \leftarrow \mathsf{pWPRF.PuncKey}(\mathsf{fk}, z^*)$, $z^* = (x^*, v)$ for some $v \leftarrow \{0,1\}^{2\lambda}$ and $\mathsf{K}^* \leftarrow \mathsf{SKE.Gen}(1^\lambda; y^*)$ such that $y^* = \mathsf{pWPRF.F}(\mathsf{fk}, z^*)$. The circuit $C[\mathsf{fk}_{z^*}, \mathsf{K}^*, X^*]$ is described as follows:

$\underline{C[\mathsf{fk}_{z^*}, \mathsf{K}^*, X^*](c, w)}$

parse $c = (c_s, x, v)$ and $X^* = (x^*, f, m_0, m_1)$
if $R(x, w) = 1$
  if $(x, v) = (x^*, v)$
    $(\hat{f}, \hat{m}) \leftarrow \mathsf{SKE.Dec}(\mathsf{K}^*, c_s)$
    if $(f = \hat{f}) \wedge (f(m_0, w) \neq f(m_1, w))$
      return $\perp$
    else return $\hat{f}(\hat{m}, w)$
  else $y \leftarrow \mathsf{pWPRF.PuncF}(\mathsf{fk}_{z^*}, (x, v))$
      $\mathsf{K} \leftarrow \mathsf{SKE.Gen}(1^\lambda; y)$
      $(\hat{f}, \hat{m}) \leftarrow \mathsf{SKE.Dec}(\mathsf{K}, c_s)$
      if $(x = x^*) \wedge (f = \hat{f}) \wedge (f(m_0, w) \neq f(m_1, w))$
        return $\perp$
      else return $\hat{f}(\hat{m}, w)$
else
  return $\perp$

It is easy to follow that the circuits $C[\mathsf{fk}, X^*], C[\mathsf{fk}_{z^*}, \mathsf{K}^*, X^*]$ compute the same function. Suppose, $(\bar{c} = (\bar{c}_s, \bar{x}, \bar{v}), \bar{w})$ is any arbitrary input to the circuits. If $z^* = (x^*, v) \neq (\bar{x}, \bar{v})$ then by the correctness of PuncF we have $\mathsf{pWPRF.F}(\mathsf{fk}, (\bar{x}, \bar{v})) = \mathsf{pWPRF.PuncF}(\mathsf{fk}_{z^*}, (\bar{x}, \bar{v}))$ and hence the circuits computes the same function. On the other hand, if $z^* = (\bar{x}, \bar{v})$, then both circuits use K^* as the SKE decryption key. By the extractability property of $e\mathcal{O}$ (Def. 13), we have

$$|\Pr[\mathsf{EG}_3] - \Pr[\mathsf{EG}_4]| = \mathsf{Adv}_{\mathcal{D}}^{e\mathcal{O}}(\lambda) = \mu(\lambda)$$

where μ is a negligible function of λ. If the advantage is not bounded by a negligible function of λ, then there exists an extractor $\mathcal{E}'$ which would produce an input where the two circuits differ, leading towards a contradiction as the circuits are equivalent.

<u>EGame 4 $\Rightarrow$ EGame 5</u>: EGame 5 samples y^* uniformly at random from $\mathcal{Y}$ instead of computing $y^* \leftarrow \mathsf{pWPRF.F}(\mathsf{fk}, (x^*, v))$ as in EGame 4, where $\mathcal{Y}$ is the co-domain of $\mathsf{pWPRF.F}(\mathsf{fk}, \cdot)$. Note that the probability of $z^* = (x^*, v) \in L'$ for a random $v \leftarrow \{0,1\}^{2\lambda}$ is negligible in λ. This means z^* is an eligible candidate to become a challenge query for a pWPRF-adversary. By the selective security of pWPRF, we have

$$|\Pr[\mathsf{EG}_4] - \Pr[\mathsf{EG}_5]| = \mathsf{Adv}_{\mathcal{B}_2}^{\mathsf{pWPRF}, R'}(\lambda)$$

where $\mathcal{B}_2$ is a pWPRF-adversary. We skip the reduction as it is similar to the reduction described in the transition from Game 3 to Game 4 of Th. 5.

Finally, the encryption key in EGame 5 is computed as $\mathsf{K} \leftarrow \mathsf{SKE.Gen}(1^\lambda; y^*)$ where y^* is a fresh randomness which is independent of the challenge statement x^*. Thus,the CIND security of SKE (Def. 5) guarantees that

$$|\Pr[\mathsf{EG}_5] - \tfrac{1}{2}| = \mathsf{Adv}_{\mathcal{B}_3}^{\mathsf{SKE}}(\lambda).$$

where $\mathcal{B}_3$ is an adversary of CIND security game. Combining all the probabilities, we get

$$
\begin{aligned}
\mathsf{Adv}_{\mathcal{A}}^{\mathsf{EOFWE},R}(\lambda) &= |\Pr[\mathsf{EG}_0] - \frac{1}{2}| \leq \sum_{i=0}^{4} |\Pr[\mathsf{EG}_i] - \Pr[\mathsf{EG}_{i+1}]| + |\Pr[\mathsf{EG}_5] - \frac{1}{2}| \\
&= \mathsf{Adv}_{\mathcal{A}}^{\mathsf{EGame\ 0\text{-}1}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{PRG}}(\lambda) + \mu(\lambda) \\
&\quad + \mathsf{Adv}_{\mathcal{B}_2}^{\mathsf{pWPRF},R'}(\lambda) + \mathsf{Adv}_{\mathcal{B}_3}^{\mathsf{SKE}}(\lambda) \\
&< \mathsf{Adv}_{\mathcal{A}}^{\mathsf{EGame\ 0\text{-}1}}(\lambda) + \mathsf{negl}(\lambda) \quad \text{(by the assumptions in the theorem)}
\end{aligned}
$$

Thus, $|\mathsf{Adv}_{\mathcal{A}}^{\mathsf{EOFWE},R}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{\mathsf{EGame\ 0\text{-}1}}(\lambda)| < \mathsf{negl}(\lambda)$ implies $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{EGame\ 0\text{-}1}}(\lambda) = \mathsf{Adv}_{\mathcal{A}}^{\mathsf{EOFWE},R}(\lambda)$ excluding the negligible term. Hence, by the similar arguments as in the transition from $\mathsf{EGame\ 0}$ to $\mathsf{EGame\ 1}$, we conclude that if $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{EOFWE},R}(\lambda) > \varrho(\lambda)$ for some non-negligible function ϱ, then there is an extractor $\mathcal{E}$ and a polynomial $\mathsf{p}_{\mathcal{E}}$ such that $\mathcal{E}(1^\lambda, X^*)$ outputs a witness w^* satisfying $R(x^*, w^*) = 1$ and $f(m_0, w^*) \neq f(m_1, w^*)$ with overwhelming probability and runs in time $\mathsf{p}_{\mathcal{E}}(\lambda, 1/\beta)$ where $\beta = (\mathsf{Adv}_{\mathcal{A}}^{\mathsf{EOFWE},R}(\lambda) - \varrho(\lambda))$. This completes the proof.