Offline Witness Encryption

Hamza Abusalah, Georg Fuchsbauer, and Krzysztof Pietrzak *

IST Austria
{habusalah, gfuchsbauer, pietrzak}@ist.ac.at

Abstract. Witness encryption (WE) was introduced by Garg et al. (STOC'13). A WE scheme is defined for some NP language L and lets a sender encrypt messages relative to instances x. A ciphertext for x can be decrypted using w witnessing $x \in L$, but hides the message if $x \notin L$. Garg et al. construct WE from multilinear maps and give another construction (FOCS'13) using indistinguishability obfuscation (iO) for encryption. Due to the reliance on such heavy tools, WE can currently hardly be implemented on powerful hardware and will unlikely be realizable on constrained devices like smart cards any time soon.
We construct a WE scheme where *encryption* is done by simply computing a Naor-Yung ciphertext (two CPA encryptions and a NIZK proof). To achieve this, our scheme has a setup phase, which outputs public parameters containing an obfuscated circuit (only required for decryption), two encryption keys and a common reference string (used for encryption). This setup need only be run once, and the parameters can be used for arbitrary many encryptions. Our scheme can also be turned into a *functional* WE scheme, where a message is encrypted w.r.t. a statement and a function f, and decryption with a witness w yields $f(m, w)$. Our construction is inspired by the functional encryption scheme by Garg et al. and we prove (selective) security assuming iO and statistically simulation-sound NIZK. We give a construction of the latter in bilinear groups and combining it with ElGamal encryption, our ciphertexts are of size 1.3 kB at a 128-bit security level and can be computed on a smart card.

Keywords: Witness Encryption, Indistinguishability Obfuscation, NIZK, Groth-Sahai Proofs.

1 Introduction

Witness encryption. In an encryption scheme, the receiver needs to know some secret piece of information (the secret key) to decrypt. Garg, Gentry, Sahai and Waters [GGSW13] propose the intriguing new notion of *witness encryption* (WE), where a scheme is defined for some NP language L with witness relation R: $L = \{x \,|\, \exists\, w : R(x, w) = 1\}$. The encryption algorithm takes an instance x (instead of a public key) and a message m and produces a ciphertext c. Using a witness w such that $R(x, w) = 1$, anyone can decrypt ciphertext c. Decryption is only possible if x is actually in the language and it is required that a ciphertext computed for some $x \notin L$ computationally hides the message m.

Applications. As shown in [GGSW13], from WE one can construct powerful cryptographic primitives such as identity-based encryption and attribute-based encryption [SW05] for circuits. But WE also allows for applications that were not possible before; for example, one can encrypt a message with respect to a puzzle, such that only someone who found a solution can decrypt. This puzzle can be any problem where solutions can be efficiently verified, like a crossword or Sudoku puzzle, or even the proof for some mathematical conjecture. Another application is asymmetric password-based encryption [BH15], which allows hashed passwords (for any password-hashing function already in place) to be used as public encryption keys and passwords to decrypt.

Constructing WE. Garg et al. [GGSW13] construct a WE scheme for the NP-complete language "exact set cover", which implies WE for any language $L \in$ NP via polynomial-time many-one reductions (a.k.a. Levin reductions). The security of this construction is based on a strong assumption on "approximate" multilinear maps as constructed in [GGH13a]. Subsequently, a construction of WE from indistinguishability obfuscation (iO) was given in [GGH+13b] and another one based on multilinear maps in [GLW14]. The only candidate construction of iO is also based on the approximate multilinear maps from [GGH13a].

* Research supported by ERC starting grant (259668-PSPC)

Implementing multilinear maps as required for iO or WE is computationally very expensive, but a first—though far from practical—implementation exists [AHKM14], and it is conceivable that algorithmic and hardware progress yield practical implementations in the not too distant future.

Offline witness encryption. Given that WE is not even practical on high-end machines, it seems foolish to hope for an implementation on low-end devices like smart cards. In this paper we show however that it is possible to construct a WE scheme where *encryption* is very efficient, as the entire computationally hard work can be moved to a setup phase and—to a lesser extent—to the decryption process. This setup is either run by the sender *before* she knows the instance and the message for an encryption; or it is run by a trusted party once and for all and everyone can use the same parameters. The first case is reminiscent of online/offline encryption or signatures [EGM96], except that in our case, once generated, the parameters can be used for arbitrary many "online phases".

We call this concept *offline witness encryption* and define it as a tuple of three algorithms. The setup phase (which is not present in standard WE) takes as input only a security parameter 1^λ and outputs public parameters $(pp_\mathsf{e}, pp_\mathsf{d}) \leftarrow \mathsf{Setup}(1^\lambda)$. To encrypt a message m for an instance x, one runs an encryption algorithm $c \leftarrow \mathsf{Enc}(x, m, pp_\mathsf{e})$. Such ciphertext c can then be decrypted given a witness w, i.e., for which $R(x, w) = 1$ holds, as $m = \mathsf{Dec}(c, w, pp_\mathsf{d})$. The goal of offline WE is to keep the parameters pp_e for encryption and ciphertexts small and the Enc algorithm efficient.

Applications of offline WE. In any application of witness encryption its offline variant can be used to make encryption practically efficient, if one accepts an additional setup phase. However, for applications like IBE and attribute-based encryption, as discussed in [GGSW13], system-wide parameters must be set up by a trusted party anyway. This party could therefore simply also generate the offline-WE parameters, meaning encryption can be made efficient without requiring any additional trust.

Bellare and Tung Hoang [BH15] define and construct *asymmetric* password-based encryption (A-PBE), where a hash of a password can be used as a public key to encrypt messages, which can then be decrypted using the password. Unlike its symmetric counterpart, A-PBE remains secure even when the server storing hashed passwords is compromised. In particular, they show that if hashed passwords are already deployed using an existing password-hashing function, witness encryption can be used to turn the hashed passwords into public keys.[1] The drawback of using WE is that both encryption and decryption are inefficient. Using offline WE where a trusted third party produces the system parameters in an offline phase, encryption can be made significantly more efficient, whereas decryption (and the one-time setup) remains inefficient.

The use of offline WE is therefore particularly appealing in scenarios where decryption is usually not done anyway, but ciphertexts are made public as a means of deterrent. Consider a scenario where a content provider lets subscribed users set up passwords and use them to access some content. The provider typically stores a hash of the password. In order to discourage subscribers from distributing their passwords and allowing others to access content, the provider could simply encrypt some sensitive user information (such as credit card details, etc.) under a user's hashed password and publish this ciphertext. As anyone who knows the password could decrypt, it is then in the user's interest to keep his password secret.

Our construction. Our construction, as well as its proof, is inspired by the *functional encryption* scheme by Garg et al. [GGH+13b].

The parameters required for encryption $pp_\mathsf{e} = (crs, pk_1, pk_2)$ consist of two public keys of a standard public-key encryption (PKE) scheme and a common reference string for a non-interactive

[1] Such a key consists of a pair (sa, hpw) of a salt and a hashed password $hpw = \mathsf{PH}(sa, pw)$ for a password-hashing function PH. Given a WE for the NP-language $\{(sa, \mathsf{PH}(sa, pw)) \mid pw\}$, messages are encrypted w.r.t. statements (sa, hpw) and can be decrypted using witness pw such that $hpw = \mathsf{PH}(sa, pw)$.

zero-knowledge (NIZK) proof system. The encryption $c = (x, c_1, c_2, \pi)$ of a message m for an instance x is simply a Naor-Yung [NY90, Sah99] CCA-secure encryption of the pair (x, m); that is, encryptions c_1 and c_2 of (x, m) under pk_1 and pk_2, respectively, together with a NIZK proof π showing that the two ciphertext c_1, c_2 encrypt the same message.

The setup algorithm samples two key pairs (sk_1, pk_1), (sk_2, pk_2) for the PKE scheme and a CRS for the NIZK proof system. The parameters pp_{d} required for decryption consist of the obfuscation $\tilde{D}$ of a circuit D defined as follows. On input a ciphertext $c = (x, c_1, c_2, \pi)$ and a string w, the circuit D

- checks if $R(x, w) = 1$ (i.e., w is a witness for $x \in L$);
- checks if π is a proof that c_1 and c_2 encrypt the same message; and
- if both checks pass, decrypts $(x', m) = \mathsf{PK.Dec}(sk_1, c_1)$ and outputs m if $x' = x$.

Given an (obfuscated) circuit as above, the decryption algorithm of our WE scheme simply evaluates $\tilde{D}((x, c_1, c_2, \pi), w)$, which will output the message m for any witness w with $R(x, w) = 1$.

We prove in Theorem 1 that the above is a secure offline-WE scheme (meaning that ciphertexts for $x \notin L$ computationally hide the message), assuming that the obfuscation satisfies the notion of *indistinguishability obfuscation* [BGI+01], the NIKZ is *statistically simulation-sound* [GGH+13b] and the PKE is semantically secure under chosen-plaintext attack (CPA).

Functional witness encryption. Functional witness encryption was proposed by Boyle et al. [BCP14] and its encryption algorithm takes as input a circuit f in addition to instance x and message m. A party knowing a witness w for x now does not learn m itself, but only the function $f(m, w)$. For example, x could be a labeled graph and a party knowing a t-clique in x can learn the labels of this clique (but no other labels). Indistinguishability-based security (there is also an extractability-based notion) requires that, even when $x \in L$, encryptions of (x, m_0, f_0) and (x, m_1, f_1) are indistinguishable if for all w with $R(x, w) = 1$ we have $f_0(m_0, w) = f_1(m_1, w)$.

In Sect. 4 we define an offline variant of functional witness encryption and give an instantiation by adapting the (obfuscated) decryption circuit of our OWE scheme: instead of outputting m when given a witness w, it parses m as a pair (f, m') and outputs $f(m', w)$. Encryption still consists of computing a Naor-Yung ciphertext, whereas for the scheme in [BCP14] the encryptor needs to perform iO-obfuscation.

Efficiency of encryption. In Sect. 5 we propose a concrete instantiation of the encryption algorithm used by our OWE schemes. In order to avoid random oracles, we use Groth-Sahai proofs [GS08], which are perfectly sound NIZK proofs in the standard model for languages defined over bilinear groups. They let us prove that two ElGamal ciphertexts encrypt the same message. Using ideas from [GGH+13b] and making them efficient by translating them into the bilinear-group framework, we construct a statistically simulation-sound (SSS) proof system. Under the so-called SXDH assumption (which states that the decisional Diffie-Hellman problem holds in the base groups), the encryption scheme is CPA-secure and the proof system we construct is zero-knowledge.

In our instantiation a proof consists of 28 elements from a bilinear group and is computed by using bilinear-group exponentiations (but no pairings). For a 128-bit security level, the size of the output of our encryption algorithm, comprising 2 ciphertexts and one SSS proof, is about 1.3 kB.

Handling long messages and instances. ElGamal encryption is defined over a group $\mathbb{G}$ and encrypts elements from $\mathbb{G}$; we therefore need to encode the message (x, m) into $\mathbb{G}$. Using elliptic-curve-based groups, for 128-bit security the length of an element from $\mathbb{G}$ is 256 bits, and standard encoding techniques [FJT13] allow for encoding of 128 bits into one group element, which is prohibitively small for any meaningful application.

We could of course choose a larger group such that one group element fits the entire tuple (x, m), but this would become very inefficient for large values. The encryption procedure we construct in

Sect. 5 will therefore allow to encrypt arbitrarily long messages by encrypting them block-wise. We then need to provide a proof for each 128-bit block separately; however, using some optimization, we manage to limit the growth of the ciphertext to 0.25 kB for every 128 bits of plaintext, meaning the ciphertext grows by a factor of 16 compared to the plaintext.

For offline WE (but not for its functional variant) a further optimization when handling large messages m is to use key encapsulation: when encrypting, the sender first picks a key k for a symmetric encryption scheme and generates a ciphertext $c = (c_K, c_M)$, where c_K is the WE encryption of (x, k), and c_M is the (secret-key) encryption of m under k. To decrypt (given a witness w), the receiver first decrypts c_K to learn k and then decrypts c_M to recover m.

Dealing with large instances x turns out more tricky. Instead of x we could encrypt a hash $y = H(x)$ using a collision-resistant hash function H, noting that x is input to the decryption algorithm, which can therefore check whether $y = H(x)$. However, to prove this construction secure, we require the notion of *differing-inputs obfuscation* (a.k.a. extractability obfuscation) [BGI+01, BCP14, ABG+13], which seems a much stronger assumption than indistinguishability obfuscation, as implausibility results in [GGHW14] show.

Related work. Zhandry [Zha14] proposes the notion of *reusable witness encryption*, which is similar to offline WE. Apart from being a key-encapsulation scheme (which does not generalize to FWE), the main difference is that setup outputs parameters which are used for both encryption and decryption and additionally a master decryption key (which allows for CCA-type security).

Zhandry constructs reusable WE using multilinear maps (and no obfuscation), which makes decryption more efficient than ours. Although ciphertexts in [Zha14] are short, the parameters are not, and, more importantly, encryption is less efficient than ours as it requires the evaluation of a multilinear map whose level of multilinearity is linear in the number of gates of the circuit describing the NP relation R. Efficient encryption was our main motivation for introducing offline WE and for this reason our model has separate parameters for encryption and decryption.

2 Preliminaries

2.1 Notations and Conventions

Families of circuits. A family of circuits $\{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ is of polynomial size if for some polynomial $p(\cdot)$ the size of every $C \in \mathcal{C}_\lambda$ is at most $|C| \leq p(\lambda)$.

Probabilistic algorithms. If $\mathcal{X}$ is a finite set then $x \leftarrow \mathcal{X}$ denotes the process of sampling x uniformly at random from $\mathcal{X}$. Let $\mathcal{A}$ be a probabilistic polynomial-time (PPT) algorithm; then $\Pr[y \leftarrow \mathcal{A}(x)]$ denotes the probability that $\mathcal{A}(x)$ outputs y when run on uniformly sampled coins. We let $\Pr\left[x_1 \leftarrow \mathcal{X}_1; x_2 \leftarrow \mathcal{X}_2; \ldots : \varphi(x_1, x_2, \ldots) = 1\right]$ denote the probability that the predicate φ evaluated on $(x_1, x_2, \ldots)$ is true after the ordered execution of $x_1 \leftarrow \mathcal{X}_1$, $x_2 \leftarrow \mathcal{X}_2, \ldots$

Negligible functions. A function $\nu\colon \mathbb{N} \to \mathbb{R}$ is called negligible if for every positive polynomial $p(\cdot)$, and all sufficiently large $n \in \mathbb{N}$, it holds that $\nu(n) \leq \frac{1}{p(n)}$. We write $f(\lambda) = \mathsf{negl}(\lambda)$ to mean that $f(\cdot)$ is negligible.

2.2 Public-Key Encryption

Our first ingredient is a standard public-key encryption scheme.

Definition 1 (PKE). *A public-key encryption scheme for a message space $\mathcal{M}$ is a tuple of PPT algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. Gen*, on input a security parameter 1^λ, outputs a secret/public key pair (sk, pk).* Enc*, on input a public key pk and a message $m \in \mathcal{M}$, outputs a ciphertext c using randomness $r \in \{0,1\}^{\ell_{\mathsf{PK}}(\lambda)}$. Finally,* Dec*, on input a secret key sk and a ciphertext c, outputs $m \in \mathcal{M} \cup \{\perp\}$. Furthermore we require correctness and security:*

$\underline{\mathbf{Exp}_{\mathcal{A}}^{\text{CPA-}b}(\lambda):}$
$(sk, pk) \leftarrow \mathsf{Gen}(1^\lambda)$
$(m_0, m_1, st) \leftarrow \mathcal{A}_1(1^\lambda, pk)$ // we require $|m_0| = |m_1|$
$c_b \leftarrow \mathsf{Enc}(pk, m_b)$
Return $b' \leftarrow \mathcal{A}_2(st, c_b)$

Fig. 1. CPA-security game of public-key encryption

– *Correctness: For every* $\lambda \in \mathbb{N}$, $m \in \mathcal{M}$ *we have*

$$\Pr\left[(sk, pk) \leftarrow \mathsf{Gen}(1^\lambda); c \leftarrow \mathsf{Enc}(pk, m) \ : \ \mathsf{Dec}(sk, c) = m\right] = 1 \ .$$

– *Indistinguishability under chosen-plaintext attacks (CPA): For every non-uniform PPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *in* $\mathbf{Exp}_{\mathcal{A}}^{\text{CPA-}b}(\lambda)$ *(Fig. 1) we have*

$$\left|\Pr\left[\mathbf{Exp}_{\mathcal{A}}^{\text{CPA-0}}(\lambda) = 1\right] - \Pr\left[\mathbf{Exp}_{\mathcal{A}}^{\text{CPA-1}}(\lambda) = 1\right]\right| = \mathsf{negl}(\lambda) \ .$$

2.3 Indistinguishability Obfuscation

As a consequence of the impossibility of virtual black-box obfuscation, Barak et al. [BGI+12] proposed the weaker notion of indistinguishability obfuscation (iO), which guarantees that obfuscations of equivalent functionalities are computationally indistinguishable.

Definition 2 (Indistinguishability obfuscation [BGI+12, GGH+13b]). *A uniform PPT algorithm* $i\mathcal{O}$ *is an indistinguishability obfuscator for a family of polynomial-size circuits* $\{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$*, if the following hold:*

– *For all* $\lambda \in \mathbb{N}$, $C \in \mathcal{C}_\lambda$, $x \in \{0,1\}^\lambda$: $\Pr\left[\widetilde{C} \leftarrow i\mathcal{O}(1^\lambda, C) : C(x) = \widetilde{C}(x)\right] = 1$.

– *For every non-uniform PPT adversary* $\mathcal{A}$*, there exists a negligible function* $\nu(\cdot)$ *such that for all* $C_0, C_1 \in \mathcal{C}_\lambda$ *such that* $C_0(x) = C_1(x)$ *for all* x*:*

$$\left|\Pr\left[\mathcal{A}(i\mathcal{O}(1^\lambda, C_0)) = 1\right] - \Pr\left[\mathcal{A}(i\mathcal{O}(1^\lambda, C_1)) = 1\right]\right| = \nu(\lambda) \ . \tag{1}$$

Garg et al. [GGH+13b] provide a candidate iO construction for families of polynomial-size circuits.

2.4 Statistically Simulation-Sound NIZK

A non-interactive (NI) proof system for a language $L \in \mathsf{NP}$ consists of four PPT algorithms: a *common-reference string* (CRS) generator G, which on input 1^λ outputs a CRS; a prover P, which on input a CRS, a statement y and a witness w outputs a proof; and a verifier V, which on input a CRS, a statement and a proof outputs 0 or 1.

We require a proof system that satisfies completeness, statistical soundness, and zero-knowledge (ZK). Completeness means that, on input a statement and a witness, P outputs a proof that V accepts. Statistical soundness requires that no unbounded adversary can produce a proof of a false statement. Zero-knowledge means that a proof does not reveal any information (in a computational sense) about the witness used to compute it; this is formalized by requiring the existence of a simulator $\mathsf{S} = (\mathsf{S}_1, \mathsf{S}_2)$ that can output a CRS and a proof for any statement, which are computationally indistinguishable from real ones.

A NIZK proof system is statistically simulation-sound (SSS) [GGH+13b] if no *unbounded* adversary can produce a valid proof for a statement $y' \notin L$ even when given a simulated proof for any other statement $y \neq y'$.

Definition 3 (SSS-NIZK). *A tuple of PPT algorithms* $(\mathsf{G}, \mathsf{P}, \mathsf{V}, \mathsf{S} = (\mathsf{S}_1, \mathsf{S}_2))$ *is a statistically simulation-sound non-interactive zero-knowledge (SSS-NIZK) proof system for* $L \in \mathsf{NP}$ *with witness relation* R *if the following hold:*

- *Perfect completeness: For every* (y, w) *such that* $R(y, w) = 1$*, it holds that*

$$\Pr\left[crs \leftarrow \mathsf{G}(1^\lambda)\,;\, \pi \leftarrow \mathsf{P}(crs, y, w) \;:\; \mathsf{V}(crs, y, \pi) = 1\right] = 1 \ .$$

- *Statistical soundness:*

$$\Pr\left[crs \leftarrow \mathsf{G}(1^\lambda) \;:\; \exists\,(y, \pi) \text{ s.t. } y \notin L \wedge \mathsf{V}(crs, y, \pi) = 1\right] = 0 \ .$$

- *Computational zero-knowledge: For every* (y, w) *such that* $R(y, w) = 1$*, and non-uniform PPT adversary* $\mathcal{A}$*, it holds that*

$$\begin{aligned}\big|\Pr\left[crs \leftarrow \mathsf{G}(1^\lambda); \pi \leftarrow \mathsf{P}(crs, y, w) \;:\; \mathcal{A}(crs, y, \pi) = 1\right] - \\ \Pr\left[(crs, \tau) \leftarrow \mathsf{S}_1(1^\lambda, y); \pi \leftarrow \mathsf{S}_2(crs, \tau, y) \;:\; \mathcal{A}(crs, y, \pi) = 1\right]\big| = \mathsf{negl}(\lambda) \ . \quad (2)\end{aligned}$$

- *Statistical simulation soundness: For every* y*, it holds that*

$$\Pr\left[\begin{array}{l}(crs, \tau) \leftarrow \mathsf{S}_1(1^\lambda, y); \\ \pi \leftarrow \mathsf{S}_2(crs, \tau, y)\end{array} \;:\; \begin{array}{l}\exists\,(y', \pi') \text{ s.t. } y' \neq y \wedge y' \notin L \\ \wedge\ \mathsf{V}(crs, y', \pi') = 1\end{array}\right] = 0 \ . \quad (3)$$

Garg et al. [GGH+13b] construct an SSS-NIZK scheme from any statistically sound NIZK scheme and any computationally hiding and perfectly binding non-interactive commitment scheme. In Sect. 5, we give an efficient instantiation of this, following their blueprint and using perfectly sound Groth-Sahai proofs [GS08] and ElGamal encryption as perfectly binding and computationally hiding commitment scheme. In particular, our SSS-NIZK proof system is for the following NP language.

Definition 4. *Let* $(\mathsf{PK.Gen}, \mathsf{PK.Enc}, \mathsf{PK.Dec})$ *be a public-key encryption scheme. We define the* NP *language* L_{enc} *and let* R_{enc} *denote its witness relation:*

$$L_{\mathsf{enc}} := \left\{(pk_1, pk_2, c_1, c_2) \;\middle|\; \begin{array}{r}\exists\,(x, m, r_1, r_2) \text{ s.t. } c_1 = \mathsf{PK.Enc}(pk_1, (x, m); r_1) \\ \wedge\ c_2 = \mathsf{PK.Enc}(pk_2, (x, m); r_2)\end{array}\right\} \ . \quad (4)$$

3 Offline Witness Encryption

A (standard) witness encryption scheme [GGSW13, BH15] is defined by an encryption algorithm Enc that takes a security parameter 1^λ, a statement x and a message m and outputs a ciphertext c; and a decryption algorithm Dec that on input a ciphertext c and a witness w outputs a message. Offline witness encryption allows for efficient encryption by outsourcing the resource-heavy computations to a setup phase, which is independent of the statement and message to be encrypted. There is a third algorithm Setup which on input a security parameter 1^λ outputs a pair of parameters: pp_{e}, which is used by Enc, and pp_{d}, which is used by Dec. In our formalization we follow the strengthened definition of witness encryption put forth by Bellare and Tung Hoang [BH15].

Definition 5 (Offline witness encryption). *An offline witness encryption (OWE) scheme for a language* $L \in \mathsf{NP}$ *with witness relation* $R : \mathcal{X} \times \mathcal{W} \to \{0, 1\}$ *is a tuple of PPT algorithms* $\mathsf{OWE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$ *where:*

- $(pp_{\mathsf{e}}, pp_{\mathsf{d}}) \leftarrow \mathsf{Setup}(1^\lambda)$*: On input a security parameter* 1^λ*,* Setup *outputs parameters for encryption* pp_{e} *and parameters for decryption* pp_{d}*.*
- $c \leftarrow \mathsf{Enc}(1^\lambda, x, m, pp_{\mathsf{e}})$*: On input a security parameter* 1^λ*, a string* $x \in \mathcal{X}$*, a message* $m \in \mathcal{M}$*, and encryption parameters* pp_{e}*,* Enc *outputs a ciphertext* c*.*

$\mathbf{Exp}^{\text{sel-WE-}b}_{L,\mathcal{A}}(\lambda)$:

$(x, m_0, m_1, st) \leftarrow \mathcal{A}_1(1^\lambda)$
// We require $|m_0| = |m_1|$
$(pp_\mathsf{e}, pp_\mathsf{d}) \leftarrow \mathsf{Setup}(1^\lambda)$
$c_b \leftarrow \mathsf{Enc}(1^\lambda, x, m_b, pp_\mathsf{e})$
$b' \leftarrow \mathcal{A}_2(st, c_b, pp_\mathsf{e}, pp_\mathsf{d})$
If $x \in L$, return 0
Return b'

Fig. 2. Selective-security game of WE

$\mathbf{Exp}^{\text{adp-WE-}b}_{L,\mathcal{A}}(\lambda)$:

$(pp_\mathsf{e}, pp_\mathsf{d}) \leftarrow \mathsf{Setup}(1^\lambda)$
$(x, m_0, m_1, st) \leftarrow \mathcal{A}_1(1^\lambda, pp_\mathsf{e}, pp_\mathsf{d})$
// We require $|m_0| = |m_1|$
$c_b \leftarrow \mathsf{Enc}(1^\lambda, x, m_b, pp_\mathsf{e})$
$b' \leftarrow \mathcal{A}_2(st, c_b)$
If $x \in L$, return 0
Return b'

Fig. 3. Adaptive-security game of WE

– $\mathsf{Dec}(c, w, pp_\mathsf{d}) \in \mathcal{M} \cup \{\perp\}$*: On input a ciphertext* c*, a string* $w \in \mathcal{W}$ *and decryption parameters* pp_d*,* Dec *outputs* $m \in \mathcal{M} \cup \{\perp\}$*.*

We require correctness and security:

– *Correctness: For all* $\lambda \in \mathbb{N}$*,* $(x, w) \in \mathcal{X} \times \mathcal{W}$ *such that* $R(x, w) = 1$*,* $m \in \mathcal{M}$*:*

$$\Pr\left[(pp_\mathsf{e}, pp_\mathsf{d}) \leftarrow \mathsf{Setup}(1^\lambda); c \leftarrow \mathsf{Enc}(1^\lambda, x, m, pp_\mathsf{e}) \ : \ \mathsf{Dec}(c, w, pp_\mathsf{d}) = m\right] .$$

– *Security:* OWE *is selectively secure if for every non-uniform PPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *in* $\mathbf{Exp}^{\text{sel-WE-}b}_{L,\mathcal{A}}(\lambda)$ *(Fig. 2) it holds that*

$$\left|\Pr\left[\mathbf{Exp}^{\text{sel-WE-0}}_{L,\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{sel-WE-1}}_{L,\mathcal{A}}(\lambda) = 1\right]\right| = \mathsf{negl}(\lambda) .$$

OWE *is adaptively secure if the same holds for* $\mathbf{Exp}^{\text{adp-WE-}b}_{L,\mathcal{A}}(\lambda)$ *(Fig. 3).*

We now present our construction of offline WE that we have outlined in the introduction and prove that it satisfies selective security.

Construction 1 (Offline WE). Let $\mathsf{PKE} = (\mathsf{PK.Gen}, \mathsf{PK.Enc}, \mathsf{PK.Dec})$ be a public-key encryption scheme, $\mathsf{NIZK} = (\mathsf{G}, \mathsf{P}, \mathsf{V})$ an SSS-NIZK scheme for L_enc (Def. 4), and $i\mathcal{O}$ an indistinguishability obfuscator for the family of polynomial-size circuits $\mathcal{D}_\lambda$ defined in (5) below. We construct an offline witness encryption scheme $\mathsf{OWE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$ for $L \in \mathsf{NP}$ that can be decided by a (circuit) witness relation $R : \{0,1\}^{\ell_x} \times \{0,1\}^{\ell_w} \to \{0,1\}$ as follows:

$(pp_\mathsf{e}, pp_\mathsf{d}) \leftarrow \mathsf{Setup}(1^\lambda)$: On input a security parameter 1^λ, do the following:

– $(sk_1, pk_1) \leftarrow \mathsf{PK.Gen}(1^\lambda)$ and $(sk_2, pk_2) \leftarrow \mathsf{PK.Gen}(1^\lambda)$.
– $crs \leftarrow \mathsf{NIZK.G}(1^\lambda)$.
– Construct the circuit $D_{sk_j, crs} \in \mathcal{D}_\lambda$ with $j = 1$

$D_{sk_j, crs}(c, w)$:
1: Parse c as $c = (x, c_1, c_2, \pi)$
2: If $\mathsf{NIZK.V}(crs, (pk_1, pk_2, c_1, c_2), \pi) = 1$
// Verify that π is a proof for (pk_1, pk_2, c_1, c_2).
// w.r.t. L_enc, where (pk_1, pk_2) is hardcoded.
3: $(\hat{x}, \hat{m}) := \mathsf{PK.Dec}(sk_j, c_j)$
4: If $(\hat{x} = x) \wedge R(x, w) = 1$
5: Return $\hat{m}$
6: Return $\perp$

(5)

– $\tilde{D}_{sk_1, crs} \leftarrow i\mathcal{O}(1^\lambda, D_{sk_1, crs})$ after padding $D_{sk_1, crs}$ appropriately.[2]

[2] W.l.o.g. we assume that $|D_{sk_1, crs}| = |D_{sk_2, crs}|$; otherwise we always pad to the maximum possible length.

$\underline{\mathbf{Exp}^{(i)}(\lambda) \quad // \; i \in \{0,1,2,3,4,5,6\}}$

$(x, m_0, m_1, st) \leftarrow \mathcal{A}_1(1^\lambda)$
$(sk_1, pk_1) \leftarrow \mathsf{PK.Gen}(1^\lambda)$; $(sk_2, pk_2) \leftarrow \mathsf{PK.Gen}(1^\lambda)$
$r_1, r_2 \leftarrow \{0,1\}^{\ell_{\mathsf{PK}}(\lambda)}$
If $i \in \{0,1,2,3\}$ $\quad c_1 := \mathsf{PK.Enc}(pk_1, (x, m_0); r_1)$
Elseif $i \in \{4,5,6\}$ $\quad c_1 := \mathsf{PK.Enc}(pk_1, (x, m_1); r_1)$
If $i \in \{0,1\}$ $\quad c_2 := \mathsf{PK.Enc}(pk_2, (x, m_0); r_2)$
Elseif $i \in \{2,3,4,5,6\}$ $\quad c_2 := \mathsf{PK.Enc}(pk_2, (x, m_1); r_2)$
$y := (pk_1, pk_2, c_1, c_2)$
If $i \in \{0,6\}$ $\quad crs \leftarrow \mathsf{NIZK.G}(1^\lambda)$
Elseif $i \in \{1,2,3,4,5\}$ $\quad (crs, \tau) \leftarrow \mathsf{NIZK.S}_1(1^\lambda, y)$
If $i \in \{0,1,2,5,6\}$ $\quad D := D_{sk_j, crs}$ with $j = 1$ as defined in (5)
Elseif $i \in \{3,4\}$ $\quad D := D_{sk_j, crs}$ with $j = 2$ as defined in (5)
$\tilde{D} \leftarrow i\mathcal{O}(1^\lambda, D)$
Set $pp_{\mathsf{e}} = (crs, pk_1, pk_2)$ and $pp_{\mathsf{d}} = \tilde{D}$
If $i = 0$ $\quad \pi \leftarrow \mathsf{NIZK.P}(crs, y, (x, m_0, r_1, r_2))$
Elseif $i = 6$ $\quad \pi \leftarrow \mathsf{NIZK.P}(crs, y, (x, m_1, r_1, r_2))$
Elseif $i \in \{1,2,3,4,5\}$ $\quad \pi \leftarrow \mathsf{NIZK.S}_2(crs, \tau, y)$
$c := (x, c_1, c_2, \pi)$
$b' \leftarrow \mathcal{A}_2(st, c, pp_{\mathsf{e}}, pp_{\mathsf{d}})$
If $x \in L$, return 0
Return b'

Fig. 4. Hybrid games used in the proof of Theorem 1

- Set $pp_{\mathsf{e}} = (crs, pk_1, pk_2)$ and $pp_{\mathsf{d}} = \tilde{D}_{sk_1, crs}$.
- Output $(pp_{\mathsf{e}}, pp_{\mathsf{d}})$.

$\underline{c \leftarrow \mathsf{Enc}(1^\lambda, x, m, pp_{\mathsf{e}})}$: On input a security parameter 1^λ, a string $x \in \{0,1\}^{\ell_x}$, a message $m \in \mathcal{M}$, and $pp_{\mathsf{e}} = (crs, pk_1, pk_2)$, Enc does the following:

- $r_1, r_2 \leftarrow \{0,1\}^{\ell_{\mathsf{PK}}(\lambda)}$.
- $c_1 := \mathsf{PK.Enc}(pk_1, (x, m); r_1)$ and $c_2 := \mathsf{PK.Enc}(pk_2, (x, m); r_2)$.
- $\pi \leftarrow \mathsf{NIZK.P}\big(crs, (pk_1, pk_2, c_1, c_2), (x, m, r_1, r_2)\big)$.
- Output $c := (x, c_1, c_2, \pi)$.

$\underline{\mathsf{Dec}(c, w, pp_{\mathsf{d}})}$: On input a ciphertext $c = (x, c_1, c_2, \pi)$, a string $w \in \{0,1\}^{\ell_w}$ and parameters $pp_{\mathsf{d}} = \tilde{D}_{sk_1, crs}$, Dec interprets $\tilde{D}_{sk_1, crs}$ as a circuit and outputs $m := \tilde{D}_{sk_1, crs}(c, w)$.

Theorem 1. $\mathsf{OWE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$ *in Construction 1 is a selectively-secure offline witness encryption scheme if* PKE *is a CPA-secure PKE scheme,* NIZK *an SSS-NIZK scheme, and* $i\mathcal{O}$ *an indistinguishability obfuscator for* $\mathcal{D}_\lambda$.

Proof. Assume towards contradiction that there exists a non-uniform PPT adversary $\mathcal{A}$ that distinguishes $\mathbf{Exp}^{\text{sel-WE-0}}_{L,\mathcal{A}}$ from $\mathbf{Exp}^{\text{sel-WE-1}}_{L,\mathcal{A}}$ with non-negligible probability. We reach a contradiction by first constructing a series of games $\mathbf{Exp}^{(i)}$ defined in Fig. 4, where by construction, $\mathbf{Exp}^{\text{sel-WE-0}}_{L,\mathcal{A}} = \mathbf{Exp}^{(0)}$ and $\mathbf{Exp}^{\text{sel-WE-1}}_{L,\mathcal{A}} = \mathbf{Exp}^{(6)}$, and then proving for $i = 0, 1, \ldots, 5$ that $\mathbf{Exp}^{(i)}$ and $\mathbf{Exp}^{(i+1)}$ are computationally indistinguishable.

$\mathbf{Exp}^{(1)}$ differs from $\mathbf{Exp}^{(0)}$ in that the CRS crs for the NIZK and the proof π are simulated rather than honestly generated. The zero-knowledge property of NIZK guarantees that honestly generated CRSs and proofs are indistinguishable from simulated ones by PPT adversaries.

Proposition 1. $\mathbf{Exp}^{(0)}(\lambda)$ *and* $\mathbf{Exp}^{(1)}(\lambda)$ *are computationally indistinguishable if* NIZK *is zero-knowledge.*

$\mathbf{Exp}^{(2)}$ differs from $\mathbf{Exp}^{(1)}$ in that the second ciphertext c_2 is generated as $\mathsf{PK.Enc}(pk_2, (x, \underline{m_1}))$ rather than $\mathsf{PK.Enc}(pk_2, (x, m_0))$. ($D_{sk_1,crs}$ and (π, crs) are the same as in $\mathbf{Exp}^{(1)}$.) The CPA-security of PKE for key pk_2 guarantees that this change is indistinguishable by PPT adversaries.

Proposition 2. $\mathbf{Exp}^{(1)}(\lambda)$ *and* $\mathbf{Exp}^{(2)}(\lambda)$ *are computationally indistinguishable if* PKE *is CPA-secure.*

$\mathbf{Exp}^{(3)}$ differs from $\mathbf{Exp}^{(2)}$ in that the circuit $D_{\underline{sk_2},crs}$ is obfuscated instead of $D_{sk_1,crs}$. Statistical simulation-soundness of NIZK now guarantees that $D_{sk_1,crs}$ and $D_{sk_2,crs}$ are functionally equivalent when crs is simulated for the statement $y := (pk_1, pk_2, c_1, c_2)$. It then follows from the security of $i\mathcal{O}$ that their obfuscations are computationally indistinguishable.

Proposition 3. $\mathbf{Exp}^{(2)}(\lambda)$ *and* $\mathbf{Exp}^{(3)}(\lambda)$ *are computationally indistinguishable if* NIZK *is statistically simulation-sound, and* $i\mathcal{O}$ *is secure.*

$\mathbf{Exp}^{(4)}$ differs from $\mathbf{Exp}^{(3)}$ in that the first ciphertext c_1 is generated as $\mathsf{PK.Enc}(pk_1, (x, \underline{m_1}))$ rather than $\mathsf{PK.Enc}(pk_1, (x, m_0))$. ($D_{sk_2,crs}$ and (π, crs) are the same as in $\mathbf{Exp}^{(3)}$.) Now CPA security of PKE w.r.t. pk_1 implies that this change is computationally indistinguishable.

Proposition 4. $\mathbf{Exp}^{(3)}(\lambda)$ *and* $\mathbf{Exp}^{(4)}(\lambda)$ *are computationally indistinguishable if* PKE *is CPA-secure.*

$\mathbf{Exp}^{(5)}$ differs from $\mathbf{Exp}^{(4)}$ in that $D_{\underline{sk_1},crs}$ is obfuscated rather than $D_{sk_2,crs}$. Statistical simulation soundness of NIZK together with security of $i\mathcal{O}$ implies that this change is computationally indistinguishable.

Proposition 5. $\mathbf{Exp}^{(4)}(\lambda)$ *and* $\mathbf{Exp}^{(5)}(\lambda)$ *are computationally indistinguishable if* NIZK *is statistically simulation-sound, and* $i\mathcal{O}$ *is secure.*

$\mathbf{Exp}^{(6)}$ is the original game $\mathbf{Exp}^{\text{sel-WE-1}}_{L,\mathcal{A}}$, and differs from $\mathbf{Exp}^{(5)}$ in that the CRS and NIZK proof (crs, π) are honestly generated rather than simulated. By zero-knowledge of NIZK this change is computationally indistinguishable.

Proposition 6. $\mathbf{Exp}^{(5)}(\lambda)$ *and* $\mathbf{Exp}^{(6)}(\lambda)$ *are computationally indistinguishable if* NIZK *is zero-knowledge.*

Theorem 1 follows from Propositions 1–6. We formally prove Theorem 1 in the full version [AFP15] □

4 Offline Functional Witness Encryption

Boyle et al. [BCP14] consider both extractable and indistinguishability-based notions of FWE. We consider an offline version of their indistinguishability-based notion. Here the encryption algorithm takes input an instance x and a pair (m, f) of a message and a description of a circuit f and outputs a ciphertext c. A party knowing a witness w for x now does not learn m itself, but only the function $f(m, w)$. Security requires computational indistinguishability of encryptions of $(x, (m_0, f_0))$ and $(x, (m_1, f_1))$ as long as $f_0(m_0, w) = f_1(m_1, w)$ for all w with $R(x, w) = 1$.

Definition 6 (Offline FWE). $\mathsf{OWE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$ *from Definition 5 is an offline FWE scheme if the following hold:*

- *Correctness: For all* $\lambda \in \mathbb{N}$, $(x, w) \in \mathcal{X} \times \mathcal{W}$ *such that* $R(x, w) = 1$, $m \in \mathcal{M}$*:*

$$\Pr\left[(pp_{\mathsf{e}}, pp_{\mathsf{d}}) \leftarrow \mathsf{Setup}(1^\lambda); c \leftarrow \mathsf{Enc}(1^\lambda, x, m, pp_{\mathsf{e}}); (m', f) := m \ : \ \mathsf{Dec}(c, w, pp_{\mathsf{d}}) = f(m', w)\right] .$$

$\underline{\mathbf{Exp}^{\text{sel-FWE-}b}_{L,\mathcal{A}}(\lambda):}$

$(x,(m'_0,f_0),(m'_1,f_1),st) \leftarrow \mathcal{A}_1(1^\lambda)$ // we require that $|(m'_0,f_0)| = |(m'_1,f_1)|$
$(pp_{\mathsf{e}},pp_{\mathsf{d}}) \leftarrow \mathsf{Setup}(1^\lambda)$; $c_b \leftarrow \mathsf{Enc}(1^\lambda,x,(m'_b,f_b),pp_{\mathsf{e}})$
$b' \leftarrow \mathcal{A}_2(st,c_b,pp_{\mathsf{e}},pp_{\mathsf{d}})$
If $\exists\, w$: $\big(R(x,w)=1 \wedge f_0(m'_0,w) \neq f_1(m'_1,w)\big)$, return 0
Return b'

Fig. 5. Security game of selectively-secure witness encryption

– *Security:* OWE *is selectively secure if for every non-uniform PPT adversary* $\mathcal{A}=(\mathcal{A}_1,\mathcal{A}_2)$ *in* $\mathbf{Exp}^{\text{sel-FWE-}b}_{L,\mathcal{A}}(\lambda)$ *(Fig. 5) we have*

$$\left|\Pr\left[\mathbf{Exp}^{\text{sel-FWE-0}}_{L,\mathcal{A}}(\lambda)=1\right]-\Pr\left[\mathbf{Exp}^{\text{sel-FWE-1}}_{L,\mathcal{A}}(\lambda)=1\right]\right| = \mathsf{negl}(\lambda)\ .$$

Construction 2 (Offline functional WE). This construction is defined exactly as Construction 1, except that in the definition of the decryption circuit in Eq. (5) on page 7 we replace

$$\text{Return } \hat{m}$$

with

$$\text{Parse } \hat{m} \text{ as } (\hat{m}',f) \text{ and return } f(\hat{m}',w)\ .$$

Theorem 2. *Construction 2 is a selectively-secure offline* functional *witness encryption scheme under the same assumptions as in Theorem 1.*

The proof is analogous to the proof of Theorem 1.[3]

5 Instantiating Enc

We now show how to efficiently instantiate the encryption algorithm of both our offline-WE schemes over a bilinear group and prove its security under a standard assumption (SXDH) and without recurring to random oracles. We use ElGamal encryption [ElG84] for the public-key encryption scheme and build an SSS-NIZK proof system from Groth-Sahai proofs [GS08] following the abstract blueprint for it given in [GGH+13b].

5.1 Tools

Bilinear groups. $\mathcal{G}$ is a bilinear-group generator if given a security parameter 1^λ it returns the description of a bilinear group $\Lambda=(p,\mathbb{G},\mathbb{H},\mathbb{T},e,g,h)$ where:

– $\mathbb{G}$, $\mathbb{H}$ and $\mathbb{T}$ are groups of prime order p, where p is of bit-length λ;
– $e\colon \mathbb{G}\times\mathbb{H}\to\mathbb{T}$ is a bilinear map, that is, $e(R^a,S^b)=e(R,S)^{ab}$ for all $R\in\mathbb{G}$, $S\in\mathbb{H}$, $a,b\in\mathbb{Z}_p$;
– g and h generate $\mathbb{G}$ and $\mathbb{H}$, resp., and $e(g,h)$ generates $\mathbb{T}$.

We use Type-3 bilinear groups [GPS08], in which no efficiently computable homomorphisms are assumed to exist between $\mathbb{G}$ and $\mathbb{H}$. We can therefore assume that the decisional Diffie-Hellman assumption (DDH) holds in $\mathbb{G}$, that is

$$\left|\Pr\left[\begin{matrix}\Lambda\leftarrow\mathcal{G}(1^\lambda);a,b\leftarrow\mathbb{Z}_p:\\ 1\leftarrow\mathcal{A}(\Lambda,g^a,g^b,g^{ab})\end{matrix}\right]-\Pr\left[\begin{matrix}\Lambda\leftarrow\mathcal{G}(1^\lambda);a,b,c\leftarrow\mathbb{Z}_p:\\ 1\leftarrow\mathcal{A}(\Lambda,g^a,g^b,g^c)\end{matrix}\right]\right| = \mathsf{negl}(\lambda)\ ,$$

[3] The only change to be made is in the proof of Proposition 3, which is the only time we use the fact that $\bar{x}\notin L$. In the description of $\mathcal{B}$, $\bar{m}_j$ is replaced by $(\bar{m}'_j,\bar{f}_j)$ for $j=0,1$. For Case 1 we now argue that $D_1((\bar{x},\bar{c}_1,\bar{c}_2,\pi),w) = D_2((\bar{x},\bar{c}_1,\bar{c}_2,\pi),w)$ for all π,w as follows: If $R(\bar{x},w)=0$ then both circuits output $\perp$. If $R(\bar{x},w)=1$ then by the winning condition for the security game we have $\bar{f}_0(\bar{m}'_0,w)=\bar{f}_1(\bar{m}'_1,w)$ for all w. Since $\bar{c}_1$ decrypts to $(\bar{x},(\bar{m}'_0,\bar{f}_0))$ and $\bar{c}_2$ decrypts to $(\bar{x},(\bar{m}'_1,\bar{f}_1))$, both circuits return $\bar{f}_0(\bar{m}'_0,w)$.

for any non-uniform PPT $\mathcal{A}$. We moreover assume DDH holds in $\mathbb{H}$, that is, the same holds with g replaced by h above. The SXDH assumption for a bilinear-group generator $\mathcal{G}$ is that DDH holds in both $\mathbb{G}$ and $\mathbb{H}$.

ElGamal encryption. We use ElGamal encryption to encrypt message vectors in $\mathbb{G}^\ell$, for some fixed ℓ. A secret key $\vec{x} \leftarrow \mathbb{Z}_p^\ell$ defines a public key $\vec{X} \in \mathbb{G}^\ell$ via $X_i := g^{x_i}$ for $i = 1, \ldots, \ell$. A message $\vec{M} = (M_i)_{i=1}^\ell \in \mathbb{G}^\ell$ is encrypted under $\vec{X}$ by choosing $r \leftarrow \mathbb{Z}_p^*$ and setting

$$\vec{c} = (c_1, \ldots, c_\ell, c_{\ell+1}) := ((M_i \cdot {X_i}^r)_{i=1}^\ell,\ g^r) \ . \tag{6}$$

Note that by using the same randomness for every component, we decrease ciphertext length. CPA security follows from the DDH assumption in $\mathbb{G}$ via a standard hybrid argument.

Groth-Sahai proofs. Groth-Sahai (GS) proofs [GS08] are efficient non-interactive witness-indistinguishable[4] (WI) proofs for several types of equations in bilinear groups. We only require *linear pairing-product equations* over variables $W_1, \ldots, W_n \in \mathbb{H}$, which are of the form

$$\prod_{i=1}^{n} e(A_i, \underline{W_i}) = t \ , \tag{7}$$

defined by $(A_i)_{i=1}^n \in \mathbb{G}^n$, and $t \in \mathbb{T}$. (As a convention, we always underline the variables.) GS proofs allow a prover to prove that there exists an assignment to the variables that satisfies a given set of equations. Groth-Sahai proofs are perfectly sound (meaning there do not exist proofs for an unsatisfiable set of equations). The instantiation of GS proofs we use is WI under the SXDH assumption. The cost of a proof is 2 elements from $\mathbb{H}$ per variable and 2 elements from $\mathbb{G}$ per equation.

5.2 Instantiation

Using ElGamal encryption, we encode pairs $M = (x, m)$ (that is, statement/message pairs which we encrypt in our offline-WE instantiation) as a vector of group elements from $\mathbb{G}^\ell$. We thus assume that there exists an efficiently decodable encoding Cd of pairs (x, m) into $\mathbb{G}^\ell$ [FJT13].

We now construct an SSS-NIZK proof system which allows us to prove that 2 ElGamal ciphertexts under different keys encrypt the same message M. We assume that the ciphertexts are always different from $(1, \ldots, 1)$, which for honestly generated ciphertext is the case as $c_{l+1} \neq 1$ in (6).

A CRS for this system consists of a CRS for GS proofs together with a commitment C to $\vec{1}$. An SSS-NIZK proof for the statement y: "$\vec{c}^{(1)}$ and $\vec{c}^{(2)}$ encrypt the same message" is a GS proof for the statement

$$\vec{c}^{(1)} \text{ and } \vec{c}^{(2)} \text{ encrypt the same message } \ \text{OR} \ \ C \text{ commits to } (\vec{c}^{(1)}, \vec{c}^{(2)}) \ . \tag{8}$$

Statistical soundness follows from perfect soundness of GS proofs: since C is not a commitment to $(\vec{c}^{(1)}, \vec{c}^{(2)}) \neq \vec{1}$, the first clause in (8) must hold. Zero-knowledge holds since given a statement $y = (\vec{c}^{(1)}, \vec{c}^{(2)})$ the simulator can set the value C in the CRS to a commitment to y; a proof for y can then be simulated by using use the second clause in (8). Since this is (in an information-theoretic sense) the only statement that can be simulated, statistical simulation-soundness (SSS) holds as well. We now present the details.

[4] Witness-indistinguishability for a proof system for a language L means the following: no PPT adversary that given crs chooses y, w_0, w_1 with $R(y, w_0) = R(y, w_1) = 1$ can distinguish $\pi_0 \leftarrow \mathsf{P}(crs, y, w_0)$ from $\pi_1 \leftarrow \mathsf{P}(crs, y, w_1)$.

Language. A statement for our language L_{enc} defined in Eq. (4) is of the form $(\vec{X}^{(1)}, \vec{X}^{(2)}, \vec{c}^{(1)}, \vec{c}^{(2)})$, where $\vec{X}^{(1)}, \vec{X}^{(2)} \in \mathbb{G}^{\ell}$ are ElGamal encryption keys and $\vec{c}^{(1)}, \vec{c}^{(2)} \in \mathbb{G}^{\ell+1}$ are ElGamal encryptions of the same message. Since the public keys are hard-coded in the description of $D_{sk_j,crs}(c,w)$ (defined in (5)), we need not include them in the statement. We therefore construct a proof system for the language

$$L_{pk_1,pk_2} := \left\{ (c_1,c_2) \;\middle|\; \begin{array}{r} \exists\,(M,r_1,r_2) \in \mathbb{G}^{\ell} \times (\mathbb{Z}_p^*)^r : c_1 = \mathsf{PK.Enc}(pk_1, M; r_1) \\ \wedge\; c_2 = \mathsf{PK.Enc}(pk_2, M; r_2) \end{array} \right\} ,$$

where M is an encoding of (x,m).

Commitment. We define a non-interactive commitment scheme that lets us commit to a message $\left(\vec{c}^{(1)}, \vec{c}^{(2)}\right) \in \mathbb{G}^{2\ell+2}$ as follows:

- The commitment key is $ck = \left(K_1^{(1)}, \ldots, K_{\ell+1}^{(1)}, K_1^{(2)}, \ldots, K_{\ell+1}^{(2)}\right) \leftarrow \mathbb{G}^{2\ell+2}$.
- A commitment $\mathsf{Com}\left(ck, (\vec{c}^{(1)}, \vec{c}^{(2)})\right)$ to a message $\left(\vec{c}^{(1)}, \vec{c}^{(2)}\right) \in \mathbb{G}^{2\ell+2}$ is computed by picking $r_c \leftarrow \mathbb{Z}_p$ and setting

$$\vec{C} = \left(\left(C_j^{(i)} := c_j^{(i)} \cdot (K_j^{(i)})^{r_c}\right)_{j=1\ldots\ell+1}^{i=1,2}, C' := g^{r_c} \right) .$$

A commitment can be opened by publishing the "opening" $W = h^{r_c}$, which allows verifying that $\vec{C}$ is a commitment to $(\vec{c}^{(1)}, \vec{c}^{(2)})$ by checking

$$\begin{aligned} e(C_j^{(i)} \cdot (c_j^{(i)})^{-1}, h) = e(K_j^{(i)}, W) \quad \text{for } i = 1,2 \;,\; j = 1\ldots,\ell+1 \text{ and} \\ e(C', h) = e(g, W) \;. \end{aligned}$$

This yields a perfectly binding commitment scheme for messages from $\mathbb{G}^{2\ell+2}$, and, as the commitment is an ElGamal encryption, it is computationally hiding under the DDH assumption in $\mathbb{G}$.

Using Com we now define our SSS proof system for showing that two ciphertexts as in (6) encrypt the same message $\vec{M}$.

CRS generation. A CRS is generated by computing a CRS for GS proofs $crs_{\mathrm{GS}} \leftarrow \mathsf{GS.G}(\Lambda)$, picking a commitment key $ck \leftarrow \mathbb{G}^{2\ell+2}$ computing $\vec{C} \leftarrow \mathsf{Com}(ck, (1,\ldots,1))$ and outputting $crs := (crs_{\mathrm{GS}}, ck, \vec{C})$.

Proof. We show how to prove, under CRS $(crs_{\mathrm{GS}}, (\vec{K}^{(1)}, \vec{K}^{(2)}), \vec{C})$, a statement $(\vec{c}^{(1)}, \vec{c}^{(2)}) \in L_{\vec{X}^{(1)},\vec{X}^{(2)}}$, using as witness (r_1, r_2) such that $\vec{c}^{(i)} = \left((M_j \cdot (X_j^{(i)})^{r_i})_{j=1}^{\ell}, g^{r_i}\right)$ for some $\vec{M} \in \mathbb{G}^{\ell}$. Consider the following set of linear pairing-product equations in variables $H_c, H_e, W_1, W_2, W_c \in \mathbb{H}$:

$$e(g, \underline{H_c})\, e(g, \underline{H_e}) = e(g,h) \tag{9}$$

$$e(C_j^{(i)} \cdot (c_j^{(i)})^{-1}, \underline{H_c}) = e(K_j^{(i)}, \underline{W_c}) \quad \text{for } i = 1,2 \;,\; j = 1\ldots,\ell+1 \tag{10}$$

$$e(C', \underline{H_c}) = e(g, \underline{W_c}) \tag{11}$$

$$e(c_j^{(1)} \cdot (c_j^{(2)})^{-1}, \underline{H_e}) = e(X_j^{(1)}, \underline{W_1})\, e((X_j^{(2)})^{-1}, \underline{W_2}) \quad \text{for } j = 1,\ldots,\ell \tag{12}$$

$$e(c_{\ell+1}^{(i)}, \underline{H_e}) = e(g, \underline{W_i}) \text{ for } i = 1,2 \tag{13}$$

A proof of our SSS-NIZK proof system is a (witness-indistinguishable) GS proof of satisfiability of the above equation system and is computed by using witness (r_1, r_2) and setting the variables to

$$H_c := 1 \;, \qquad H_e := h \;, \qquad W_c := 1 \;, \qquad W_1 := h^{r_1} \;, \qquad W_2 := h^{r_2} \;. \tag{14}$$

Verification. A proof π for statement $(\vec{c}^{(1)}, \vec{c}^{(2)})$ under CRS $(crs_{\mathrm{GS}}, ck, \vec{C})$ is verified by verifying the GS proof π under crs_{GS} of satisfiability of equations (9)–(13) defined by the values in $\vec{c}^{(1)}, \vec{c}^{(2)}, ck = (\vec{K}^{(1)}, \vec{K}^{(2)})$ and $\vec{C} = ((C_j^{(i)})_{j=1\ldots\ell+1}^{i=1,2}, C' := g^{r_c})$.

Completeness of our SSS NIZK proof system follows from completeness of GS proofs together with the fact that the values in (14) satisfy (9)–(13).

Soundness. Below we show that a proof of satisfiability of equations (9)–(13) proves that

$$\begin{array}{l} - \textbf{ either } \vec{c}^{(1)} \text{ and } \vec{c}^{(2)} \text{ are encryptions of the same message} \\ - \textbf{ or } \vec{C} \text{ contained in the CRS is a commitment to } (\vec{c}^{(1)}, \vec{c}^{(2)}). \end{array} \tag{15}$$

Since GS proofs are perfectly sound and an honestly generated CRS contains a commitment to $(1, \ldots, 1)$, which is a valid statement, a valid proof shows that the "either" clause above is satisfied, thus $(\vec{c}^{(1)}, \vec{c}^{(2)}) \in L_{pk_1, pk_2}$. We now show (15).

- Eq. (9) proves that either $H_c \neq 1$ or $H_e \neq 1$; since $e(g,1)\,e(g,1) \neq e(g,h)$.
- If $H_c \neq 1$ then (10)–(11) prove that $(C_1^{(1)}, \ldots, C_{\ell+1}^{(1)}, C_1^{(2)}, \ldots, C_{\ell+1}^{(2)}, C')$ commits to $(c_1^{(1)}, \ldots, c_{\ell+1}^{(1)}, c_1^{(2)} \ldots, c_{\ell+1}^{(2)})$:
  Let $\eta, \omega \in \mathbb{Z}_p$, $\eta \neq 0$ (since $H_c \neq 1$), be such that $H_c = h^\eta$ and $W_c = h^\omega$. From (11) we have $C' = g^{\omega/\eta}$, whereas the equations in (10) yield $C_j^{(i)} \cdot (c_j^{(i)})^{-1} = (K_j^{(i)})^{\omega/\eta}$, thus $C_j^{(i)} = c_j^{(i)} \cdot (K_j^{(i)})^{\omega/\eta}$, which together means that $(C_1^{(1)}, \ldots, C_{\ell+1}^{(2)}, C')$ is a commitment to $(c_1^{(1)}, \ldots, c_{\ell+1}^{(2)})$ with randomness $r_c = \omega/\eta$.
- If $H_e \neq 1$ then with $\eta \neq 0, \omega_1$ and ω_2 such that $H_e = h^\eta$ and $W_i = h^{\omega_i}$ the equations in (13) yield that $c_{\ell+1}^{(i)} = g^{\omega_i/\eta}$, for $i = 1, 2$. Set $r_i := \omega_i/\eta$ and let $m_j^{(i)}$ be (the unique values) such that $c_j^{(i)} = g^{m_j^{(i)}} \cdot (X_j^{(i)})^{r_i}$. Then the equations in (12) yield $c_j^{(1)} \cdot (c_j^{(2)})^{-1} = (X_j^{(1)})^{r_1} \cdot (X_j^{(2)})^{-r_2}$, thus $g^{m_j^{(1)}} = g^{m_j^{(2)}}$ for all $j = 1, \ldots, \ell$, meaning $\vec{c}^{(1)}$ and $\vec{c}^{(2)}$ encrypt the same message.

Simulation. Given a statement $(\vec{c}^{(1)}, \vec{c}^{(2)})$, the simulator sets up the CRS by choosing $r_c \leftarrow \mathbb{Z}_p$ and setting $\vec{C} := \mathsf{Com}(ck, (\vec{c}^{(1)}, \vec{c}^{(2)}); r_c)$. It simulates a proof for statement $(\vec{c}^{(1)}, \vec{c}^{(2)}) \in L_{pk_1, pk_2}$ by computing a GS proof for equations (9)–(13) by instantiating the variables as

$$H_c := h\ , \qquad H_e := 1\ , \qquad W_c := h^{r_c}\ , \qquad W_1 := 1\ , \qquad W_2 := 1\ .$$

Since the commitment in the CRS is hiding under DDH in $\mathbb{G}$, and since GS proofs are witness-indistinguishable under SXDH, this simulation is also indistinguishable under SXDH (which implies DDH in $\mathbb{G}$). Statistical simulation-soundness holds, since once the CRS is set up, $(\vec{c}^{(1)}, \vec{c}^{(2)})$ is the only statement for which a proof using the 2nd clause in (15) can be computed. Any other proof must use the first clause, meaning the statement must be in the language.

5.3 Cost of an Encryption

In standard implementations of bilinear groups for 128-bit security, $\mathbb{G}$ elements are of size 256 bits and $\mathbb{H}$ elements are of size 512 bits. Let ℓ be such that pairs (x, m) are of size $< 128 \cdot \ell$ bits, that is, they can be mapped to $\mathbb{G}^\ell$.

An encryption in our WE scheme then consists of two ElGamal ciphertexts (each in $\mathbb{G}^{\ell+1}$) and a GS proof with 5 variables in $\mathbb{H}$ (requiring 10 elements from $\mathbb{H}$) and $3\ell+6$ linear equations (requiring $6\ell+12$ elements from $\mathbb{G}$). Computing an ElGamal encryption requires $\ell+1$ exponentiations and ℓ group operations in $\mathbb{G}$. The 2 elements from $\mathbb{H}$ required for each variable require 2 exponentiations and one group operation in $\mathbb{H}$. The 2 elements from $\mathbb{G}$ required for each equation are computed using together 4 exponentiations and 2 group operations in $\mathbb{G}$.

With the above instantiation the output of Enc is in $\mathbb{G}^{8\ell+14} \times \mathbb{H}^{10}$. If two group elements suffice to encode pairs (x, m) then one encryption has ≈ 1.6 kB. For every 128-bit increase of the message length, the encryption only grows by 8 elements from $\mathbb{G}$, that is 0.25 kB.

Acknowledgements. We would like to thank Ilan Komargodski and Mark Zhandry for clarifying the definition of reusable witness encryption, and Alex Bredariol Grilo for pointing out that a generic instantiation of our offline witness encryption construction requires *perfect* SSS NIZK.

References

ABG+13. Prabhanjan Ananth, Dan Boneh, Sanjam Garg, Amit Sahai, and Mark Zhandry. Differing-inputs obfuscation and applications. *IACR Cryptology ePrint Archive*, 2013:689, 2013.

AFP15. Hamza Abusalah, Georg Fuchsbauer, and Krzysztof Pietrzak. Offline witness encryption. Cryptology ePrint Archive, Report 2015/838, 2015. http://eprint.iacr.org/.

AHKM14. Daniel Apon, Yan Huang, Jonathan Katz, and Alex J. Malozemoff. Implementing cryptographic program obfuscation. Cryptology ePrint Archive, Report 2014/779, 2014. http://eprint.iacr.org/.

BCP14. Elette Boyle, Kai-Min Chung, and Rafael Pass. On extractability obfuscation. In Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 52–73. Springer, Heidelberg, February 2014.

BGI+01. Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 1–18. Springer, Heidelberg, August 2001.

BGI+12. Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. *J. ACM*, 59(2):6:1–6:48, May 2012.

BH15. Mihir Bellare and Viet Tung Hoang. Adaptive witness encryption and asymmetric password-based cryptography. In Jonathan Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 308–331, 2015.

EGM96. Shimon Even, Oded Goldreich, and Silvio Micali. On-line/off-line digital signatures. *Journal of Cryptology*, 9(1):35–67, 1996.

ElG84. Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and David Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 10–18. Springer, Heidelberg, August 1984.

FJT13. Pierre-Alain Fouque, Antoine Joux, and Mehdi Tibouchi. Injective encodings to elliptic curves. In Colin Boyd and Leonie Simpson, editors, *ACISP 13*, volume 7959 of *LNCS*, pages 203–218. Springer, Heidelberg, July 2013.

GGH13a. Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 1–17. Springer, Heidelberg, May 2013.

GGH+13b. Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49. IEEE Computer Society Press, October 2013.

GGHW14. Sanjam Garg, Craig Gentry, Shai Halevi, and Daniel Wichs. On the implausibility of differing-inputs obfuscation and extractable witness encryption with auxiliary input. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 518–535. Springer, Heidelberg, August 2014.

GGSW13. Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 467–476. ACM Press, June 2013.

GLW14. Craig Gentry, Allison B. Lewko, and Brent Waters. Witness encryption from instance independent assumptions. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 426–443. Springer, Heidelberg, August 2014.

GPS08. Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

GS08. Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, Heidelberg, April 2008.

NY90. Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd ACM STOC*, pages 427–437. ACM Press, May 1990.

Sah99. Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th FOCS*, pages 543–553. IEEE Computer Society Press, October 1999.

SW05. Amit Sahai and Brent R. Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 457–473. Springer, Heidelberg, May 2005.

Zha14. Mark Zhandry. How to avoid obfuscation using witness PRFs. Cryptology ePrint Archive, Report 2014/301, 2014. http://eprint.iacr.org/2014/301.

A Proofs

Below, we assume that the adversary $\mathcal{A}$ is deterministic, which is without loss of generality as we always can fix $\mathcal{A}$'s random coins to some value maximizing its advantage. As $\mathcal{A}$ has zero advantage if the x it initially outputs is in L, we can further assume w.l.o.g. that the x initially output by $\mathcal{A}$ is never in L.

A.1 Proof of Proposition 1

Proof. Assume towards contradiction that there exists a polynomial $p(\cdot)$ such that for infinitely many λ

$$\left|\Pr\left[\mathbf{Exp}^{(0)}(\lambda)=1\right]-\Pr\left[\mathbf{Exp}^{(1)}(\lambda)=1\right]\right|\geq\tfrac{1}{p(\lambda)}\ .$$

We use $\mathcal{A}$ to construct a non-uniform PPT adversary $\mathcal{B}$ against the zero-knowledge security of NIZK (cf.(2)) as follows:[5]

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

- $(x,m_0,m_1,st)\leftarrow\mathcal{A}_1(1^\lambda)$.
- $(sk_1,pk_1)\leftarrow\mathsf{PK.Gen}(1^\lambda)$ and $(sk_2,pk_2)\leftarrow\mathsf{PK.Gen}(1^\lambda)$.
- $r_1,r_2\leftarrow\{0,1\}^{\ell_{\mathsf{PK}}(\lambda)}$.
- $c_1:=\mathsf{PK.Enc}(pk_1,(x,m_0);r_1)$ and $c_2:=\mathsf{PK.Enc}(pk_2,(x,m_0);r_2)$.
- Set $y=(pk_1,pk_2,c_1,c_2)$ and $w=(x,m_0,r_1,r_2)$ and note that $R_{\mathsf{enc}}(y,w)=1$.
- Submit (y,w) to the zero-knowledge game of (2) to obtain either
  - An honest (crs^*,π^*): $crs^*\leftarrow\mathsf{NIZK.G}(1^\lambda)$ and $\pi^*\leftarrow\mathsf{NIZK.P}(crs^*,y,w)$, or
  - A simulated (crs^*,π^*): $(crs^*,\tau)\leftarrow\mathsf{NIZK.S}_1(1^\lambda,y)$, and $\pi^*\leftarrow\mathsf{NIZK.S}_2(crs^*,\tau,y)$
- Set $\pi=\pi^*$ and $crs=crs^*$.
- Construct $D:=D_{sk_j,crs}$ with $j=1$ as defined in (5).
- $\tilde{D}\leftarrow i\mathcal{O}(1^\lambda,D)$.
- Set $pp_{\mathsf{e}}=(crs,pk_1,pk_2)$, $pp_{\mathsf{d}}=\tilde{D}$, and $c=(x,c_1,c_2,\pi)$.
- Output $b'\leftarrow\mathcal{A}_2(st,c,pp_{\mathsf{e}},pp_{\mathsf{d}})$.

By construction, if (crs^*,π^*) is generated honestly then $\mathcal{B}$ simulates $\mathbf{Exp}^{(0)}$, and if (crs^*,π^*) is simulated then $\mathcal{B}$ simulates $\mathbf{Exp}^{(1)}$. Therefore, for infinitely many λ it holds that

$$\tfrac{1}{p(\lambda)}\leq\left|\Pr\left[\mathbf{Exp}^{(0)}(\lambda)=1\right]-\Pr\left[\mathbf{Exp}^{(1)}(\lambda)=1\right]\right|=\left|\Pr\left[\begin{array}{l}crs\leftarrow\mathsf{G}(1^\lambda);\\ \pi\leftarrow\mathsf{P}(crs,y,w)\end{array}:\mathcal{B}(crs,y,\pi)=1\right]-\Pr\left[\begin{array}{l}(crs,\tau)\leftarrow\mathsf{S}_1(1^\lambda,y);\\ \pi\leftarrow\mathsf{S}_2(crs,\tau,y)\end{array}:\mathcal{B}(crs,y,\pi)=1\right]\right|\ .$$

We therefore reach a contradiction to the zero-knowledge security of NIZK, and conclude that

$$\left|\Pr\left[\mathbf{Exp}^{(0)}(\lambda)=1\right]-\Pr\left[\mathbf{Exp}^{(1)}(\lambda)=1\right]\right|=\mathsf{negl}(\lambda)\ .$$

A.2 Proof of Proposition 2

Proof. Assume towards contradiction that there exists a polynomial $p(\cdot)$ such that for infinitely many λ

$$\left|\Pr\left[\mathbf{Exp}^{(1)}(\lambda)=1\right]-\Pr\left[\mathbf{Exp}^{(2)}(\lambda)=1\right]\right|\geq\tfrac{1}{p(\lambda)}\ .$$

We use $\mathcal{A}$ to construct a non-uniform PPT adversary $\mathcal{B}=(\mathcal{B}_1,\mathcal{B}_2)$ for the CPA security game $\mathbf{Exp}_{\mathcal{B}}^{\text{CPA-}b}(\lambda)$ of PKE (cf. Fig. 1) as follows:

$\underline{\mathcal{B}_1(1^\lambda,pk)}$:

- $(x,m_0,m_1,st_{\mathcal{A}})\leftarrow\mathcal{A}_1(1^\lambda)$.
- $(sk_1,pk_1)\leftarrow\mathsf{PK.Gen}(1^\lambda)$ and set $pk_2=pk$.
- $r_1\leftarrow\{0,1\}^{\ell_{\mathsf{PK}}(\lambda)}$.
- $c_1:=\mathsf{PK.Enc}(pk_1,(x,m_0);r_1)$.
- Set $m_0'=(x,m_0)$, $m_1'=(x,m_1)$, and $st=(sk_1,pk_1,c_1,r_1,x,m_0,m_1,st_{\mathcal{A}})$.
- Output (m_0',m_1',st).

[5] We even break a *weaker* definition of zero-knowledge where (2) is only required to hold for pairs (y,w) with $R(y,w)=1$ output by $\mathcal{A}$ rather than for *any* such pair.

$\underline{\mathcal{B}_2(st, c'_b)}$:

- Set $c_2 = c'_b$ and $y = (pk_1, pk_2, c_1, c_2)$.
- $(crs, \tau) \leftarrow \mathsf{NIZK.S}_1(1^\lambda, y)$.
- $\pi \leftarrow \mathsf{NIZK.S}_2(crs, \tau, y)$.
- Construct $D := D_{sk_j, crs}$ with $j = 1$ as defined in (5).
- $\tilde{D} \leftarrow i\mathcal{O}(1^\lambda, D)$.
- Set $pp_{\mathsf{e}} = (crs, pk_1, pk_2)$, $pp_{\mathsf{d}} = \tilde{D}$, and $c = (x, c_1, c_2, \pi)$.
- Output $b' \leftarrow \mathcal{A}_2(st_{\mathcal{A}}, c, pp_{\mathsf{e}}, pp_{\mathsf{d}})$.

By construction, if $c'_b \leftarrow \mathsf{PK.Enc}(pk, m'_0)$ then $\mathcal{B}$ simulates $\mathbf{Exp}^{(1)}$, and if $c'_b \leftarrow \mathsf{PK.Enc}(pk, m'_1)$ then $\mathcal{B}$ simulates $\mathbf{Exp}^{(2)}$. Therefore, for infinitely many λ, it holds that

$$\frac{1}{p(\lambda)} \leq \left| \Pr\left[\mathbf{Exp}^{(1)}(\lambda) = 1\right] - \Pr\left[\mathbf{Exp}^{(2)}(\lambda) = 1\right] \right| = \left| \Pr\left[\mathbf{Exp}^{\text{CPA-0}}_{\mathcal{B}}(\lambda) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{CPA-1}}_{\mathcal{B}}(\lambda) = 1\right] \right| .$$

We therefore reach a contradiction to the CPA security of PKE, and conclude that

$$\left| \Pr\left[\mathbf{Exp}^{(1)}(\lambda) = 1\right] - \Pr\left[\mathbf{Exp}^{(2)}(\lambda) = 1\right] \right| = \mathsf{negl}(\lambda) .$$

A.3 Proof of Proposition 3

Proof. Assume towards contradiction that there exists a polynomial $p(\cdot)$ such that for infinitely many λ

$$\left| \Pr\left[\mathbf{Exp}^{(2)}(\lambda) = 1\right] - \Pr\left[\mathbf{Exp}^{(3)}(\lambda) = 1\right] \right| \geq \frac{1}{p(\lambda)} .$$

Then we use $\mathcal{A}$ to construct a non-uniform PPT adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ against the indistinguishability security of $i\mathcal{O}$ (cf. (1)) as follows:[6]

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

- $(\bar{x}, \bar{m}_0, \bar{m}_1, st) \leftarrow \mathcal{A}_1(1^\lambda)$.
- $(sk_1, pk_1) \leftarrow \mathsf{PK.Gen}(1^\lambda)$ and $(sk_2, pk_2) \leftarrow \mathsf{PK.Gen}(1^\lambda)$.
- $r_1, r_2 \leftarrow \{0,1\}^{\ell_{\mathsf{PK}}(\lambda)}$.
- Set $\bar{c}_1 = \mathsf{PK.Enc}(pk_1, (\bar{x}, \bar{m}_0); r_1)$ and $\bar{c}_2 = \mathsf{PK.Enc}(pk_2, (\bar{x}, \bar{m}_1); r_2)$.
- Set $\bar{y} = (pk_1, pk_2, \bar{c}_1, \bar{c}_2)$.
- $(crs, \tau) \leftarrow \mathsf{NIZK.S}_1(1^\lambda, \bar{y})$.
- $\bar{\pi} \leftarrow \mathsf{NIZK.S}_2(crs, \tau, \bar{y})$.
- Construct $D_j := D_{sk_j, crs}$ for $j = 1, 2$ as defined in (5).
- Submit (D_1, D_2) to the $i\mathcal{O}$ challenger to obtain $\tilde{D} \leftarrow i\mathcal{O}(1^\lambda, D_j)$.
- Set $pp_{\mathsf{e}} = (crs, pk_1, pk_2)$, $pp_{\mathsf{d}} = \tilde{D}$, and $\bar{c} = (\bar{x}, \bar{c}_1, \bar{c}_2, \bar{\pi})$.
- Output $b' \leftarrow \mathcal{A}_2(st_{\mathcal{A}}, \bar{c}, pp_{\mathsf{e}}, pp_{\mathsf{d}})$.

By construction, if $\tilde{D} \leftarrow i\mathcal{O}(1^\lambda, D_1)$ then $\mathcal{B}$ simulates $\mathbf{Exp}^{(2)}$, and if $\tilde{D} \leftarrow i\mathcal{O}(1^\lambda, D_2)$ then $\mathcal{B}$ simulates $\mathbf{Exp}^{(3)}$. Therefore, for infinitely many λ, it holds that

$$\frac{1}{p(\lambda)} \leq \left| \Pr\left[\mathbf{Exp}^{(2)}(\lambda) = 1\right] - \Pr\left[\mathbf{Exp}^{(3)}(\lambda) = 1\right] \right| = \left| \Pr\left[\mathcal{B}(i\mathcal{O}(1^\lambda, D_1)) = 1\right] - \Pr\left[\mathcal{B}(i\mathcal{O}(1^\lambda, D_2)) = 1\right] \right| . \quad (16)$$

We show that statistical simulation soundness of NIZK implies $D_1 \equiv D_2$. Let $((x, c_1, c_2, \pi), w)$ be an arbitrary input. We distinguish the following cases and show that D_1 and D_2 have the same output.

[6] We actually break a *weaker* definition of iO where (1) is only required to hold for pairs (C_0, C_1) with $C_0 \equiv C_1$ output by $\mathcal{A}$ rather than for *any* such pair.

Case 1 $x = \bar{x}$, $c_1 = \bar{c}_1$ and $c_2 = \bar{c}_2$: Since $\bar{x} \notin L$, we have $R(x, w) = 0$, thus both D_1 and D_2 return $\perp$ because of the 2nd check in line 4.

Case 2 $x \neq \bar{x}$, $c_1 = \bar{c}_1$ and $c_2 = \bar{c}_2$: By correctness of PKE, circuit D_j computes $(\bar{x}, \bar{m}_j)$ in line 3 and both return $\perp$ because $\bar{x} \neq x$ in line 4.

Case 3 $c_1 \neq \bar{c}_1$ or $c_2 \neq \bar{c}_2$: If $\mathsf{PK.Dec}(sk_0, c_1) = \mathsf{PK.Dec}(sk_1, c_2)$ then both D_1 and D_2 have the same behavior (as the circuits only differ in line 3). If the decryptions differ then by correctness of PKE we have $y := (pk_1, pk_2, c_1, c_2) \notin L_{\mathsf{enc}}$. Since $y \neq \bar{y}$ (for which crs was simulated by $\mathsf{NIZK.S_1}$), by SSS of NIZK we have $\mathsf{NIZK.V}(crs, y, \pi) = 0$, which means that both D_1 and D_2 return $\perp$ because of the check in line 2.

We thus have $D_1 \equiv D_2$ which together with (16) contradicts the security of $i\mathcal{O}$. We conclude that

$$\left| \Pr\left[\mathbf{Exp}^{(2)}(\lambda) = 1\right] - \Pr\left[\mathbf{Exp}^{(3)}(\lambda) = 1\right] \right| = \mathsf{negl}(\lambda) \ .$$

A.4 Proofs of Propositions 4, 5 and 6

The proofs of Propositions 4, 5 and 6 are analogous to those of Propositions 2, 3 and 1, respectively.