A preliminary version of this paper appears in the proceedings of PKC 2015. This is the full version, appearing as IACR ePrint Archive Report 2013/704.

Adaptive Witness Encryption and Asymmetric Password-based Cryptography

Mihir Bellare[1] Viet Tung Hoang[2]

February 11, 2015

Abstract

We show by counter-example that the soundness security requirement for witness encryption given by Garg, Gentry, Sahai and Waters (STOC 2013) does not suffice for the security of their own applications. We introduce adaptively-sound (AS) witness encryption to fill the gap. We then introduce asymmetric password-based encryption (A-PBE). This offers gains over classical, symmetric password-based encryption in the face of attacks that compromise servers to recover hashed passwords. We distinguish between invasive A-PBE schemes (they introduce new password-based key-derivation functions) and non-invasive ones (they can use existing, deployed password-based key-derivation functions). We give simple and efficient invasive A-PBE schemes and use AS-secure witness encryption to give non-invasive A-PBE schemes.

[1] Department of Computer Science & Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. Email: `mihir@eng.ucsd.edu`. URL: `http://cseweb.ucsd.edu/~mihir/`. Supported in part by NSF grants CNS-1116800 and CNS-1228890.

[2] Department of Computer Science, University of Maryland, College Park, and Department of Computer Science, Georgetown University, 37th and O Streets, NW, Washington, DC 20057, USA. Email: `tvhoang@umd.edu`. URL: `http://csiflabs.cs.ucdavis.edu/~tvhoang/`. Supported in part by NSF award CNS-1223623. Part of the work was done while Hoang was working at UCSD and supported in part by NSF grants CNS-1116800, and CNS-1228890.

Contents

1 Introduction

This paper introduces (1) witness encryption with adaptive soundness security and (2) asymmetric password-based encryption (A-PBE). We show how to use (1) to achieve (2) as well as other goals.

The problem. The security of Internet communication remains ubiquitously based on client passwords. Standards such as the widely implemented PKCS#5 —equivalently, RFC 2898 [33]— specify password-based encryption (PBE). From the client password pw, one derives a hashed password $hpw = \mathsf{PH}(sa, pw)$, where sa is a random, user-specific public salt, and PH is a deterministic password-hashing function. (In the standards, $\mathsf{PH}(sa, pw) = H^t(sa|pw)$ where t is an iteration count and H^t denotes the t-fold iteration of cryptographic hash function H.) The server holds hpw while the client holds (sa, pw). Now the server will encrypt under hpw using any symmetric encryption scheme, for example CBC-AES. The client can recompute hpw from (sa, pw) and decrypt using this key.

This classical form of PBE is *symmetric*: encryption and decryption are both done under the same key hpw. But this means that anyone who knows hpw can decrypt. This is a serious vulnerability in practice because of server compromise leading to exposure of hashed passwords. The Heartbleed attack of April 2014, allowing an attacker to read large chunks of server memory that can contain sensitive client information including hashed passwords, is a recent and prominent instance. Other high-profile attacks that compromised servers to expose client information include Target (December 2013), Adobe (October 2013), LinkedIn (June 2012), RSA (March 2011), Sony (2011) and TJ Maxx (2007). According to CNBC, there were over 600 breaches in 2013 alone.

We emphasize that the problem here is not the possibility of password-recovery via a dictionary attack based on the hashed password. The problem is that S-PBE (symmetric PBE) is vulnerable *even if the password is well chosen to resist dictionary attack.* This is because possession of the hashed password is already and directly enough to decrypt any prior communications. So under S-PBE, even well-chosen passwords do not provide security in the face of server compromise.

A-PBE. We propose asymmetric password-based encryption (A-PBE). Here, encryption is done under the hashed password hpw, decryption is done under the password pw, and *possession of* hpw *does not allow decryption.* This offers significantly higher security in the face of the most important attack, namely server compromise exposing the hashed password hpw.

This paper initiates a foundational treatment of A-PBE including definitions and both "invasive" and "non-invasive" schemes. At first it may appear that definitionally A-PBE is just like PKE and brings nothing new, but this is not true. Not just is security based on passwords, but in practice users pick related passwords, for example varying a base password by appending the name of the website, resulting in encryption under related keys. Our definition extends the S-PBE framework of [9]. Our security model explicitly considers encryption under *multiple* passwords, assumed to be individually unpredictable —otherwise security is not possible— but arbitrarily related to each other.

We give two proven-secure A-PBE schemes that we call APBE1 and APBE2. Their attributes are summarized in Fig. 1. APBE1 is simple, natural and as efficient as possible, but what we call invasive, is that it specifies its own password-hashing function PH. APBE2 is non-invasive, meaning able to use any, given password-hashing function. In particular it can work with in-use, standardized password hashing functions such as PKCS#5 [33] or bcrypt [34]. If one has the flexibility of changing PH and the associated password hashes then the first solution is preferable. The second solution may be easier to deploy in the face of the legacy constraint of millions of existing, PKCS#5 hashed passwords.

Scheme	Achieves	Invasive	Assumptions
APBE1	Secure A-PBE	Yes	PKE, RIP-secure hash
APBE2	Secure A-PBE	No	AS-secure WE, RIP-secure password hash with large stretch
			XS-secure WE, ROW-secure password hash with arbitrary stretch

Figure 1: **Our A-PBE schemes.** Both achieve our notion of security for related, unpredictable passwords. APBE1 has a dedicated password hash (invasive) while APBE2 can work with an arbitrary, legacy one (non-invasive). The first analysis of APBE2 assumes the password hash has large stretch, a restriction dropped in the second analysis under a stronger form of WE.

APBE1. We specify and analyze the following simple and natural scheme for A-PBE that we call APBE1. PH, given sa, pw, applies to them a deterministic function EX to derive a string r, uses this as coin tosses for a key-generation algorithm $\mathsf{PKE.Kg}$ of some standard PKE scheme to get (pk, sk), and outputs $hpw = pk$ as the hashed password. Encryption is under the encryption algorithm $\mathsf{PKE.Enc}$ of the PKE scheme keyed with $hpw = pk$. Since PH is deterministic, decryption under (sa, pw) can re-execute PH to get (sk, pk) and then use sk to decrypt under PKE.

A natural choice for EX is a randomness extractor [32] with seed sa. But recall that we require A-PBE to be secure even under multiple, related passwords. To achieve this, outputs of EX must be independent even if the input passwords are related, and an extractor does not guarantee this. Indeed it is not possible for this to be true information theoretically, meaning if the "independence" is required to be statistical. We instead target computational independence of the outputs of EX. We define an appropriate security goal for EX that we call related-input pseudorandomness (RIP) [29] and show that this together with security of the base PKE scheme suffices for the security of the A-PBE scheme. In practice, EX can be efficiently instantiated via HMAC [5].

Non-invasive A-PBE. APBE1 prescribes its own password-hashing algorithm under which the hashed password hpw is a public key of some existing PKE scheme. In current practice, however, the hashed password is derived via the iterated hashing password-hash function of PKCS#5 [33] or alternatives such as bcrypt [34]. Right now millions of passwords are in use with these particular password-hashing functions. In the face of this legacy constraint, deployment of A-PBE would be eased by a scheme that could encrypt under an existing, given hashed password, regardless of its form. We ask whether such non-invasive A-PBE is achievable.

This turns out to be challenging, even in principle, let alone in practice. In all known PKE schemes, the secret and public keys have very specific structure and are related in very particular ways. How can we encrypt asymmetrically with the public key being just an arbitrary hash of the secret key?

The answer is witness encryption (WE), introduced by Garg, Gentry, Sahai and Waters (GGSW) [19]. We will use WE to achieve non-invasive A-PBE. For this purpose, however, we will need WE schemes satisfying an extension of the soundness security notion of GGSW [19] that we introduce and call adaptive soundness security. We define and achieve WE with adaptive soundness and apply it to achieve non-invasive A-PBE as we now discuss.

SS-secure witness encryption. In a WE scheme [19] for a language $L \in \mathbf{NP}$, the encryption function $\mathsf{WE.Enc}$ takes a unary representation 1^λ of the security parameter $\lambda \in \mathbb{N}$, a string $x \in$

$\{0,1\}^*$ and a message m to return a ciphertext c. If $x \in L$ then decryption is possible given a witness w for the membership of x in L. If $x \notin L$ then the message remains private given the ciphertext. The soundness security (SS) requirement of GGSW [19] formalized the latter by asking that for any PT adversary A, any $x \notin L$ and any equal-length messages m_0, m_1, there is a negligible function ν such that $\Pr[A(\mathsf{WE.Enc}(1^\lambda, x, m_1)) = 1] - \Pr[A(\mathsf{WE.Enc}(1^\lambda, x, m_0)) = 1] \leq \nu(\lambda)$ for all $\lambda \in \mathbb{N}$.

AS-secure witness encryption. Our (new) adaptive soundness (AS) requirement lets the adversary A, on input 1^λ, pick and return x, m_0, m_1 to the game. The latter picks a random challenge bit b and returns ciphertext $\mathsf{WE.Enc}(1^\lambda, x, m_b)$ to A, who now responds with a guess b' as to the value of b. The AS-advantage of A is defined as the probability that $(b = b')$ and $x \notin L$. We require that any PT A have negligible advantage. We note that due to the check that $x \notin L$, our game may not be polynomial time but this does not hinder our applications.

It may at first seem that adaptivity does not add strength, since soundness security already quantifies over all x, m_0, m_1. But in fact we show that AS is strictly stronger than SS. Namely we show in Proposition 3.2 that AS always implies SS but SS does not necessarily imply AS. That is, any WE scheme that is AS secure is SS secure, but there exist WE schemes that are SS secure and not AS secure. Intuitively, the reason AS is strictly stronger is that SS does not allow x, m_0, m_1 to depend on λ. Our separation result modifies a SS-secure WE scheme to misbehave when $|x| \geq f(\lambda)$ for a certain poly-logarithmic function f of the security parameter. SS is preserved because for each x only finitely many values of λ trigger the anomaly. The proof that AS is violated uses the fact that $\mathbf{NP} \subseteq \mathbf{EXP}$, the constructed adversary nonetheless being polynomial time.

Having strengthened the goal, we must revisit achievability. GGHRSW [17] give an elegant and conceptually simple construction of SS-secure WE from indistinguishability obfuscation (iO). In Theorem 3.3 we show that the same construction achieves the stronger AS goal. Recent work has provided constructions of iO improved both along the assumptions and efficiency fronts [15, 3, 23, 2], leading to corresponding improvements for AS-secure WE. Thus AS-secure WE can be achieved without loss of efficiency or added assumptions compared to SS-secure WE.

APBE2. Our APBE2 scheme lets L be the $\mathbf{NP}$ language of pairs $(sa, \mathsf{PH}(sa, pw))$ over the choices of sa, pw, the witness being pw. A-PBE encryption of m using the hashed password as the public key will be AS-secure witness encryption of m under $x = (sa, hpw)$. Decryption will use the witness pw.

This solution is non-invasive, as it does not prescribe or require any particular design for PH. Rather, it takes PH as given, and shows how to encrypt with public key the hashed password obtained from PH. In this way, PH can in particular be the iterated hash design of the PKCS#5 standard [33] that already underlies millions of usages of passwords, or any other practical, legacy design. Of course, for security, we will need to make an assumption about the security of PH, but that is very different from prescribing its design. Our assumption is the same RIP security as discussed above. We note that this assumption is already, even if implicitly, made in practice for the security of in-use S-PBE, where the hashed passwords are the keys, and is shown by [9] to hold for PKCS#5 in the ROM, so it is a natural and reasonable assumption.

SS revisited. GGSW [19, 20] present constructions of PKE, IBE and ABE schemes from witness encryption, claiming that these constructions are secure assuming soundness security of the WE scheme. The need for adaptive security of our A-PBE scheme leads to the natural question of why we need a stronger condition than GGSW [19, 20]. The answer is that they need it too. We point out that the theorems of GGSW [19, 20] claiming security of their applications under SS are incorrect, and that SS does not in fact suffice for the security of their schemes. We do this by presenting counter-examples (cf. Section 4). Taking their PRG-based PKE construction as a representative example, we provide a WE scheme which satisfies SS yet, if used in their construction, the resulting PKE scheme will provide no security at all. We then show that the gap can be filled by using AS. Namely, we show that their PKE scheme is secure if the underlying WE scheme is AS secure and the PRG is secure. Analogous results hold for GGSW's applications to IBE and ABE. Intuitively, the weakness of SS that compromises the applications of GGSW [19, 20] is that a WE scheme may satisfy SS yet behave totally insecurely, for example returning the message in the clear, when $|x| = \lambda$. But in applications, x will have length related to λ, so SS is not enough. AS does not have this weakness because x can depend on λ.

Better security for APBE2. Define the stretch of a password-hashing function as the difference between its output length and input length, and denote it by s. Our result of Theorem 5.1 proving the security of APBE2 requires that 2^{-s} is negligible, meaning the output length is somewhat more than the input length. This captures situations in which passwords are, say 12-character ASCII strings (input length is 78-bit) and the password hashing function is iterated SHA1 (output length is 160-bit). However, when passwords are longer, say 24-character, then passwords should offer more security. To fill this gap we offer a second analysis of the security of APBE2 that removes the restriction on the stretch, allowing it now to be arbitrary. For this purpose we strengthen the assumption on the WE scheme from AS to a notion of adaptive extractability we call XS. As a side benefit, the prior assumption on the password hashing function (RIP security, asking that password hashes are pseudorandom) is reduced to ROW security, asking merely that the password hashing function is one way.

XS is an adaptive variant of the notion of extractability from GKPVZ [26]. XS asks that, given an adversary violating the security of the encryption under $x \in \{0,1\}^*$, one can extract a witness w for the membership of $x \in L$, even when x depends on the security parameter. We show that XS implies AS and also that XS-secure WE can be achieved based on extractable (aka. differing-input) obfuscations [4, 13, 1].

Some works [14, 18] cast doubts on the achievability of extractable witness encryption or extractable iO with *arbitrary auxiliary inputs*. Our result however requires a very particular auxiliary input and the attacks in these works do not apply.

A-PBE as PKE. The standard model for public-key encryption (PKE) is that the user (receiver) publishes a public encryption key and stores the corresponding secret key securely. In practice, however, the secret key is often not stored in computer memory but instead derived from a password stored in human memory. Reasons this is advantageous include *security* and *mobility*. Computer-stored keys are vulnerable to exfiltration by malware. Meanwhile, users tend to have numerous devices including cellphones and tablets on which they want to decrypt. They may also use web-based services such as gmail on untrusted client machines. Passwords are more flexible and secure than stored keys in such settings.

A-PBE captures this more real-world PKE model. Our definitions allow us to evaluate security in the setting of actual use, namely when secret keys are possibly correlated passwords. Our schemes provide solutions with provable guarantees. We note that A-PBE is the model of the recently proposed gmail end-to-end encryption system, evidencing practical relevance of the goal.

Password-based signatures. Beyond A-PBE, we view this paper as initiating a study of asymmetric password-based cryptography. In this light we also introduce and treat password-based signatures with both invasive and non-invasive solutions to mirror the case of A-PBE.

Password-based authentication is currently done using a MAC keyed by the hashed password. It is thus subject to the same weakness as S-PBE, namely that compromise of the server through

Heartbleed or other attacks leads to compromise of hashed passwords, resulting in compromise of the authentication. In the password-based signatures we suggest, one signs under the password pw and verifies under the hashed password $hpw = \mathsf{PH}(sa, pw)$. Possession of the hashed password does not compromise security.

We can give a simple solution analogous to the one for A-PBE, namely apply a RIP function EX to the password and salt to get coins; run a key-generation of a standard digital signature scheme on these to get a signing key and verification key; set the password hash to the verification key; to sign given the password, re-generate the signing and verifying keys and sign under the former. This, however is invasive, prescribing its own password-hashing function. It is a good choice if one has the flexibility of implementing a new password hashing function, but as discussed above, deployment in the face of legacy PKCS#5 password hashes motivates asking whether a non-invasive solution, meaning one that can utilize any given password hashing function, is possible. As with A-PBE, this is a much more challenging question. We can show how to obtain a non-invasive password-based signature scheme by using key-versatile signatures [8]. The latter are effectively witness signatures meeting strong simulatability and extractability conditions [16, 8] and allow us to obtain password-based signatures analogous to how we obtained A-PBE from WE. The only assumption needed on the password hashing function PH is that it is one-way.

Discussion and GGSW updates. A good definition for WE should have two properties: (1) *Usability*, meaning it suffices to prove security of applications, and (2) *Achievability*, meaning proposed and natural constructions, which in this case mainly means the iO-based one of GGHRSW [17], can be shown to meet the definition. Our AS definition has both properties, making it viable. We have shown that SS lacked the usability property.

Here we have referred to the original GGSW STOC paper [19] and the corresponding original full ePrint version [20]. Subsequent to seeing prior versions of our paper, the GGSW authors updated their paper on ePrint [21, 22]. They acknowledge the gap we found. They also propose their own, modified definitions in an attempt to fill this gap.

These updated definitions remain, from our perspective, problematic. We showed that their first proposed definition, which we call SS2 [21], is unachievable. (Because the negligible function is not allowed to depend on the adversary. See Appendix A.) We communicated this to the authors. They then updated SS2 to SS3 [22]. But we explain in Appendix A that SS3 has limitations with regard to achievability. While one might of course propose still further modifications to their definition it is not clear why this is a productive route for the community in the face of the fact that, with AS, we have —and had prior to the GGSW updates— a definition that provides both usability and achievability.

Recently KNY [31] gave a definition, that we call SS5, in the quantifier style of SS1, SS2 and SS3. We discuss it also in Appendix A where we show that it is unachievable. (Because, like SS2, the negligible function doesn't depend on the adversary.)

These developments are an indication that neither the gap we find, nor the AS definition we propose to fill it, are trivial, that quantifier-based definitions are error-prone, and that our counter-examples for SS remain important to understand and guide definitional choices. Demonstrating the last, beyond [21, 22], further work subsequent to ours, and definitionally influenced by ours, includes [24].

We believe the idea of witness encryption is important and useful and we view our work as advancing its cause. Precision in definitions, proofs and details is particularly important in our field because we claim proven security. Reaching such precision can require iteration and definitional adjustments and increments, and our work, in this vein, helps towards greater impact and clarity for the area of witness encryption.

2 Preliminaries

Notation. We denote the size of a finite set X by $|X|$, the number of coordinates of a vector $\mathbf{x}$ by $|\mathbf{x}|$, and the length of a string $x \in \{0,1\}^*$ by $|x|$. We let ε denote the empty string. By $x\|y$ we denote the concatenation of strings x, y. If X is a finite set, we let $x \leftarrow\!\!\$\, X$ denote picking an element of X uniformly at random and assigning it to x. Algorithms may be randomized unless otherwise indicated. Running time is worst case. "PT" stands for "polynomial-time," whether for randomized algorithms or deterministic ones. If A is an algorithm, we let $y \leftarrow A(x_1, \ldots; r)$ denote running A with random coins r on inputs $x_1, \ldots$ and assigning the output to y. We let $y \leftarrow\!\!\$\, A(x_1, \ldots)$ be the resulting of picking r at random and letting $y \leftarrow A(x_1, \ldots; r)$. We say that $f : \mathbb{N} \to \mathbb{R}$ is negligible if for every positive polynomial p, there exists $n_p \in \mathbb{N}$ such that $f(n) < 1/p(n)$ for all $n > n_p$. An adversary is an algorithm or a tuple of algorithms.

Games. We use the code based game playing framework of [10]. For an example of a game see Fig. 2. By $\mathrm{G}^A(\lambda) \Rightarrow y$ we denote the event that the execution of game G with adversary A and security parameter λ results in the game returning y. We abbreviate $\mathrm{G}^A(\lambda) \Rightarrow \mathsf{true}$ by $\mathrm{G}^A(\lambda)$, the occurrence of this event meaning that A wins the game.

Unpredictability. Let $A = (A_1, \ldots)$ be a tuple of algorithms where A_1, on input the unary representation 1^λ of the security parameter $\lambda \in \mathbb{N}$, returns a vector $\mathbf{pw}$. Let $\mathrm{Guess}_A(\lambda)$ denote the maximum, over all i, pw, of $\Pr[\mathbf{pw}[i] = pw]$, the probability over $\mathbf{pw} \leftarrow\!\!\$\, A_1(\lambda)$. We say that A is *unpredictable* if the function $\mathrm{Guess}_A(\cdot)$ is negligible.

3 Adaptive Witness Encryption

We begin by recalling the notion of witness encryption of GGSW [19] and their soundness security requirement. We then give a different security notion called adaptive soundness. We show that it is strictly stronger than the original, which means we must address achieving it. We show that it is achievable via indistinguishability obfuscation.

NP relations. For R: $\{0,1\}^* \times \{0,1\}^* \to \{\mathsf{true}, \mathsf{false}\}$, we let $\mathsf{R}(x) = \{\, w : \mathsf{R}(x, w)\,\}$ be the *witness set* of $x \in \{0,1\}^*$. We say R is an **NP**-relation if it is computable in PT and there is a polynomial $\mathsf{R.wl}$: $\mathbb{N} \to \mathbb{N}$, called the witness length of R, such that $\mathsf{R}(x) \subseteq \{0,1\}^{\mathsf{R.wl}(|x|)}$ for all $x \in \{0,1\}^*$. We let $\mathcal{L}(\mathsf{R}) = \{\, x : \mathsf{R}(x) \neq \emptyset \,\} \in \mathbf{NP}$ be the language defined by R.

WE syntax and correctness. A witness encryption (WE) scheme WE for $L = \mathcal{L}(\mathsf{R})$ defines a pair of PT algorithms $\mathsf{WE.Enc}, \mathsf{WE.Dec}$. Algorithm $\mathsf{WE.Enc}$ takes as input the unary representation 1^λ of a security parameter $\lambda \in \mathbb{N}$, a string $x \in \{0,1\}^*$, and a message $m \in \{0,1\}^*$, and outputs a ciphertext c. Algorithm $\mathsf{WE.Dec}$ takes as input a string w and a ciphertext c, and outputs $m \in \{0,1\}^* \cup \{\bot\}$. Correctness requires that $\mathsf{WE.Dec}(w, \mathsf{WE.Enc}(1^\lambda, x, m)) = m$ for all $\lambda \in \mathbb{N}$, all $x \in L$, all $w \in \mathsf{R}(x)$ and all $m \in \{0,1\}^*$.

Soundness security. The soundness security (SS) condition of GGSW [19] says that for any PT adversary A, any $x \in \{0,1\}^* \setminus L$ and any equal-length $m_0, m_1 \in \{0,1\}^*$, there is a negligible function ν such that for all $\lambda \in \mathbb{N}$ we have

$$\Pr[A(\mathsf{WE.Enc}(1^\lambda, x, m_1)) = 1] - \Pr[A(\mathsf{WE.Enc}(1^\lambda, x, m_0)) = 1] < \nu(\lambda)\ . \tag{1}$$

In the following, it is useful to let $\mathsf{Adv}^{\mathsf{ss}}_{\mathsf{WE},L,x,m_0,m_1,A}(\lambda)$ denote the probability difference in Equation (1). Then the soundness condition can be succinctly and equivalently stated as follows: WE is SS[L]-secure if for any PT adversary A, any $x \in \{0,1\}^* \setminus L$ and any equal-length $m_0, m_1 \in \{0,1\}^*$, the

Game $\mathrm{AS}^{A}_{\mathsf{WE},L}(\lambda)$
$(x, m_0, m_1, \mathrm{St}) \leftarrow\$ A(1^\lambda)$; $b \leftarrow\$ \{0,1\}$; $c \leftarrow\$ \mathsf{WE}(1^\lambda, x, m_b)$; $b' \leftarrow\$ A(\mathrm{St}, c)$
Return $((b = b') \wedge (x \notin L))$

Figure 2: Game AS defining adaptive soundness of witness encryption scheme WE.

$\mathsf{WE}_f.\mathsf{Enc}(1^\lambda, x, m)$	$\mathsf{WE}_f.\mathsf{Dec}(w, c)$
If $\|x\| \geq f(\lambda)$ then return $(0, m)$	$(b, t) \leftarrow c$
Else return $(1, \mathsf{WE}.\mathsf{Enc}(1^\lambda, x, m))$	If $b = 0$ then return t else return $\mathsf{WE}.\mathsf{Dec}(w, t)$

Figure 3: Witness encryption scheme WE_f for $L \in \mathbf{NP}$, derived from $\mathsf{WE} \in \mathsf{SS}[L]$ and a PT-computable function $f : \mathbb{N} \to \mathbb{N}$.

---

function $\mathsf{Adv}^{\mathsf{ss}}_{\mathsf{WE},L,x,m_0,m_1,A}(\cdot)$ is negligible. It is convenient, in order to succinctly and precisely express relations between notions, to let $\mathsf{SS}[L]$ denote the set of all correct witness encryption schemes that are SS[L]-secure.

Adaptive soundness. Our security definition associates to witness encryption scheme WE, language $L \in \mathbf{NP}$, adversary A and $\lambda \in \mathbb{N}$ the game $\mathrm{AS}^{A}_{\mathsf{WE},L}(\lambda)$ of Fig. 2. Here the adversary, on input 1^λ, produces instance x, messages m_0, m_1, and state information St. It is required that $|m_0| = |m_1|$. The game picks a random challenge bit b and computes a ciphertext c via $\mathsf{WE}.\mathsf{Enc}(1^\lambda, x, m_b)$. The adversary is now given c, along with its state information St, and outputs a prediction b' for b. The game returns true if the prediction is correct, meaning $b = b'$, and also if $x \notin L$. We let $\mathsf{Adv}^{\mathsf{as}}_{\mathsf{WE},L,A}(\lambda) = 2\Pr[\mathrm{AS}^{A}_{\mathsf{WE},L}(\lambda)] - 1$. We say that WE has adaptive soundness security for L, or is AS[L]-secure, if for every PT A the function $\mathsf{Adv}^{\mathsf{as}}_{\mathsf{WE},L,A}(\cdot)$ is negligible. We let $\mathsf{AS}[L]$ denote the set of all correct witness encryption schemes that are AS[L]-secure.

Due to the check that $x \notin L$, our game does not necessarily run in PT. This, however, will not preclude applicability. The difference between AS and SS is that in the former, x, m_0, m_1 can depend on the security parameter and on each other. Given that SS quantifies over all x, m_0, m_1, this may not at first appear to make any difference. But we will see that it does and that AS is strictly stronger than SS.

AS is a game-based definition while SS is phrased in a more "quantifier-based" style that mimics the soundness condition in interactive proofs [28]. The game-based AS notion is better suited for applications because the latter are also underlain by game-based definitions. Indeed we'll see that SS does not suffice for applications.

A useful transform. In several proofs, we'll employ the following transform. Given a WE scheme $\mathsf{WE} \in \mathsf{SS}[L]$ and a PT function $f\colon \mathbb{N} \to \mathbb{N}$, our transform returns another WE scheme WE_f. The constructed scheme, formally specified in Fig. 3, misbehaves, returning the message in the clear, when $|x| \geq f(\lambda)$, and otherwise behaves like WE. The following says that if f is chosen to satisfy certain conditions then SS[L]-security is preserved, meaning $\mathsf{WE}_f \in \mathsf{SS}[L]$. In our uses of the transform we will exploit the fact that WE_f will fail to have other security properties or lead to failure of applications that use it.

Lemma 3.1 Let $L \in \mathbf{NP}$ and $\mathsf{WE} \in \mathsf{SS}[L]$. Let $f : \mathbb{N} \to \mathbb{N}$ be a non-decreasing, PT-computable function such that $\lim_{\lambda\to\infty} f(\lambda) = \infty$. Consider witness encryption scheme WE_f derived from WE and f as shown in Fig. 3. Then $\mathsf{WE}_f \in \mathsf{SS}[L]$.

Proof: Let A be a PT adversary. Let $x \in \{0,1\}^* \setminus L$ and let $m_0, m_1 \in \{0,1\}^*$ have equal length. Let PT adversary B, on input ciphertext c, return $b' \leftarrow A((1,c))$. Let $S(x) = \{\lambda \in \mathbb{N} : f(\lambda) \le |x|\}$. Then for all $\lambda \in \mathbb{N} \setminus S(x)$ we have $\mathsf{Adv}^{\mathsf{ss}}_{\mathsf{WE},L,x,m_0,m_1,B}(\lambda) = \mathsf{Adv}^{\mathsf{ss}}_{\mathsf{WE}_f,L,x,m_0,m_1,A}(\lambda)$. The assumption that $\mathsf{WE} \in \mathsf{SS}[L]$ means that $\mathsf{Adv}^{\mathsf{ss}}_{\mathsf{WE},L,x,m_0,m_1,B}(\cdot)$ is negligible. But the assumptions on f mean that the set $S(x)$ is finite. Consequently, the function $\mathsf{Adv}^{\mathsf{ss}}_{\mathsf{WE}_f,L,x,m_0,m_1,A}(\cdot)$ is negligible as well. ∎

Relations. We show that adaptive soundness implies soundness but not vice versa, meaning adaptive soundness is a strictly stronger requirement.

Proposition 3.2 Let $L \in \mathbf{NP}$. Then: (1) $\mathsf{AS}[L] \subseteq \mathsf{SS}[L]$, and (2) If $\{0,1\}^* \setminus L$ is infinite and $\mathsf{SS}[L] \neq \emptyset$ then $\mathsf{SS}[L] \not\subseteq \mathsf{AS}[L]$.

Claim (1) above says that any witness encryption scheme WE that is AS[L]-secure is also SS[L]-secure. Claim (2) says that the converse is not true. Namely, there is a witness encryption scheme WE such that WE is SS[L]-secure but not AS[L]-secure. This separation assumes some SS[L]-secure witness encryption scheme exists, for otherwise the claim is moot. It also assumes that the complement of L is not trivial, meaning is infinite, which is true if L is $\mathbf{NP}$-complete and $\mathbf{P} \neq \mathbf{NP}$, hence is not a strong assumption.

Proof of Proposition 3.2: For part (1), assume we are given WE that is AS[L]-secure. We want to show that WE is SS[L]-secure. Referring to the definition of soundness security, let A be a PT adversary, let $x \in \{0,1\}^* \setminus L$ and let $m_0, m_1 \in \{0,1\}^*$ have equal length. We want to show that the function $\mathsf{Adv}^{\mathsf{ss}}_{\mathsf{WE},L,x,m_0,m_1,A}(\cdot)$ is negligible. We define the adversary B_{x,m_0,m_1} as follows: Let $B_{x,m_0,m_1}(1^\lambda)$ return $(x, m_0, m_1, \varepsilon)$ and let $B_{x,m_0,m_1}(t,c)$ return $b' \leftarrow\!\!\$\; A(c)$. Here, B_{x,m_0,m_1} has x, m_0, m_1 hardwired in its code, and, in its first stage, it returns them, along with $\mathrm{St} = \varepsilon$ as state information. In its second stage, it simply runs A. Note that even though B_{x,m_0,m_1} has hardwired information, this information is finite and not dependent on the security parameter, so the hardwiring does not require non-uniformity. Now it is easy to see that for all $\lambda \in \mathbb{N}$ we have $\mathsf{Adv}^{\mathsf{as}}_{\mathsf{WE},L,B_{x,m_0,m_1}}(\lambda) = \mathsf{Adv}^{\mathsf{ss}}_{\mathsf{WE},L,x,m_0,m_1,A}(\lambda)$. The assumption that WE is AS[L]-secure means that $\mathsf{Adv}^{\mathsf{as}}_{\mathsf{WE},L,B_{x,m_0,m_1}}(\cdot)$ is negligible, hence so is $\mathsf{Adv}^{\mathsf{ss}}_{\mathsf{WE},L,x,m_0,m_1,A}(\cdot)$, as desired.

For part (2), the assumption $\mathsf{SS}[L] \neq \emptyset$ means there is some $\mathsf{WE} \in \mathsf{SS}[L]$. By way of Lemma 3.1, we can modify it to $\mathsf{WE}_f \in \mathsf{SS}[L]$ as specified in Fig. 3, where $f : \mathbb{N} \to \mathbb{N}$ is some non-decreasing, PT-computable function such that $\lim_{\lambda\to\infty} f(\lambda) = \infty$. Now we want to present an attacker A violating AS[L]-security of WE_f. The difficulty is that A needs to find $x \notin L$ of length $f(\lambda)$, but $L \in \mathbf{NP}$ and A must be PT. We will exploit the fact that $\mathbf{NP} \subseteq \mathbf{EXP}$ and pick f to be a poly-logarithmic function related to the exponential time to decide L, so that if there exists an $x \notin L$ of length $f(\lambda)$ then A can find it by exhaustive search in PT. Our assumption that the complement of L is infinite means that A succeeds on infinitely many values of λ.

Proceeding to the details, since $L \in \mathbf{NP} \subseteq \mathbf{EXP}$, there is a constant $d \ge 1$ and a deterministic algorithm M such that for every $x \in \{0,1\}^*$, we have $M(x) = 1$ if and only if $x \in L$, and M's running time is $\mathcal{O}(2^{|x|^d})$. Define f by $f(\lambda) = \lfloor \lg^{1/d}(\lambda) \rfloor$ for all $\lambda \in \mathbb{N}$. Let $\mathsf{WE} \in \mathsf{SS}[L]$ and let WE_f be the witness encryption scheme derived from WE and f as specified in Fig. 3. By Lemma 3.1, $\mathsf{WE}_f \in \mathsf{SS}[L]$. Now we show that $\mathsf{WE}_f \notin \mathsf{AS}[L]$. Let $m_0, m_1 \in \{0,1\}^*$ be arbitrary, distinct, equal-length messages. Consider the following adversary A:

$$\begin{array}{|l|}\hline \underline{\text{GAME IO}_{\mathsf{P}}^{A}(\lambda)} \\ (C_0, C_1, \mathrm{St}) \leftarrow\!\!\$\, A(1^\lambda)\ ;\ b \leftarrow\!\!\$\, \{0,1\}\ ;\ c \leftarrow\!\!\$\, \mathsf{P.Ob}(1^\lambda, C_b) \\ b' \leftarrow\!\!\$\, A(\mathrm{St}, c)\ ;\ \text{Return } (b = b') \wedge (C_0 \equiv C_1) \\ \hline \end{array}$$

Figure 4: Game IO defining security of an indistinguishability obfuscator P.

$$\begin{array}{l|l} \underline{A(1^\lambda)} & \underline{A(t,c)} \\ k \leftarrow f(\lambda)\ ;\ x \leftarrow 0^k & (b,m) \leftarrow c \\ \text{For all } s \in \{0,1\}^k \text{ do} & \text{If } ((b=0) \wedge (m = m_1)) \text{ then return } 1 \\ \quad \text{If } (M(s) \neq 1) \text{ then } x \leftarrow s & \text{Return } 0 \\ \text{Return } (x, m_0, m_1, \varepsilon) & \end{array}$$

Each execution of M takes time $\mathcal{O}(2^{k^d}) = \mathcal{O}(\lambda)$. The For loop goes through all $s \in \{0,1\}^k$ in lexicographic order and thus M is executed at most $2^k \leq \lambda$ times. So A is PT. For any $\lambda \in \mathbb{N}$, if $\{0,1\}^{f(\lambda)} \setminus L \neq \emptyset$ then $\mathsf{Adv}^{\mathsf{as}}_{\mathsf{WE}_f,L,A}(\lambda) = 1$. Since $\{0,1\}^* \setminus L$ is infinite, f is non-decreasing, and $\lim_{t\to\infty} f(t) = \infty$, there are infinitely many values λ such that $\mathsf{Adv}^{\mathsf{as}}_{\mathsf{WE}_f,L,A}(\lambda) = 1$, and thus $\mathsf{WE}_f \notin \mathsf{AS}[L]$, as claimed. ▌

Indistinguishability obfuscation. We say that two circuits C_0 and C_1 are *functionally equivalent*, denoted $C_0 \equiv C_1$, if they have the same size, the same number n of inputs, and $C_0(x) = C_1(x)$ for every input $x \in \{0,1\}^n$. An obfuscator P defines PT algorithms $\mathsf{P.Ob}, \mathsf{P.Ev}$. Algorithm $\mathsf{P.Ob}$ takes as input the unary representation 1^λ of a security parameter λ and a circuit C, and outputs a string c. Algorithm $\mathsf{P.Ev}$ takes as input strings c, x and returns $y \in \{0,1\}^* \cup \{\perp\}$. We require that for any circuit C, any input x, and any $\lambda \in \mathbb{N}$, it holds that $\mathsf{P.Ev}(x, \mathsf{P.Ob}(1^\lambda, C)) = C(x)$. We say that P is iO-secure if $\mathsf{Adv}^{\mathsf{io}}_{\mathsf{P},A}(\lambda) = 2\Pr[\mathrm{IO}^A_{\mathsf{P}}(\lambda)] - 1$ is negligible for every PT adversary A, where game IO is defined at Fig. 4. This definition is slightly different from the notion in [4, 17]—the adversary is non-uniform and must produce functionally equivalent circuits C_0 and C_1—but the former definition is implied by the latter.

Achieving AS-security. Our AS security notion is strictly stronger than the SS one of GGSW [19], but we'll show that the iO-based WE scheme of GGHRSW [17] is AS-secure. Proceeding to the details, let R be an **NP**-relation. For each $x, m \in \{0,1\}^*$, let $R_{x,m}$ be a circuit that, on input $w \in \{0,1\}^{\mathsf{R.wl}(|x|)}$, returns m if $\mathsf{R}(x,w)$ and returns $0^{|m|}$ otherwise. Let P be an indistinguishability obfuscator, defining a PT obfuscation algorithm $\mathsf{P.Ob}$ and a PT evaluation algorithm $\mathsf{P.Ev}$. We define WE scheme $\mathsf{WE_R[P]}$ as follows: algorithm $\mathsf{WE_R[P].Enc}(1^\lambda, x, m)$ returns $c \leftarrow\!\!\$\, \mathsf{P.Ob}(1^\lambda, R_{x,m})$; and algorithm $\mathsf{WE_R[P].Dec}(w, c)$ returns $m \leftarrow\!\!\$\, \mathsf{P.Ev}(w, c)$.

Theorem 3.3 Let R be an **NP**-relation and let $L = \mathcal{L}(\mathsf{R})$. Let P be an indistinguishability obfuscator. Construct $\mathsf{WE_R[P]}$ as above. If P is iO-secure then $\mathsf{WE_R[P]} \in \mathsf{AS}[L]$.

Proof: Let A be a PT adversary attacking the AS[L]-security of $\mathsf{WE_R[P]}$. Wlog, assume that A produces distinct m_0 and m_1. Note that $R_{x,m_0} \equiv R_{x,m_1}$ if and only if $x \notin L$. Consider the following PT adversary B attacking iO-security of P:

$$\begin{array}{l|l} \underline{B(1^\lambda)} & \underline{B(\mathrm{St}, c)} \\ (x, m_0, m_1, \mathrm{St}) \leftarrow\!\!\$\, A(1^\lambda)\ ;\ \text{Return } (R_{x,m_0}, R_{x,m_1}, \mathrm{St}) & b' \leftarrow\!\!\$\, A(\mathrm{St}, c)\ ;\ \text{Return } b' \end{array}$$

Game $\mathrm{PRG}_G^A(\lambda)$	Game $\mathrm{INDCPA}_{\mathsf{PKE}}^A(\lambda)$
$s \leftarrow_\$ \{0,1\}^\lambda$; $x_1 \leftarrow G(s)$	$(pk, sk) \leftarrow_\$ \mathsf{PKE.Kg}(1^\lambda)$; $b \leftarrow_\$ \{0,1\}$
$x_0 \leftarrow_\$ \{0,1\}^{\ell(\lambda)}$; $b \leftarrow_\$ \{0,1\}$	$b' \leftarrow_\$ A^{\mathrm{LR}}(1^\lambda, pk)$; Return $(b = b')$
$b' \leftarrow_\$ A(1^\lambda, x_b)$; Return $(b = b')$	$\underline{\mathrm{LR}(m_0, m_1)}$
	$c \leftarrow_\$ \mathsf{PKE.Enc}(pk, m_b)$; Return c

Figure 5: **Left:** Game PRG defining security of a pseudorandom generator G. Here $\ell : \mathbb{N} \to \mathbb{N}$ is the expansion factor of G. **Right:** Game INDCPA defining INDCPA security of a PKE scheme PKE. For each oracle query, the messages $m_0, m_1 \in \{0,1\}^*$ must have the same length.

$\mathsf{PKE.Kg}(1^\lambda)$	$\mathsf{PKE.Enc}(pk, m)$	$\mathsf{PKE.Dec}(sk, c)$
$sk \leftarrow_\$ \{0,1\}^\lambda$; $x \leftarrow G(sk)$	$(\lambda, x) \leftarrow pk$	Return $\overline{\mathsf{WE}}.\mathsf{Dec}(c, sk)$
$pk \leftarrow (\lambda, x)$; Return (pk, sk)	Return $\overline{\mathsf{WE}}.\mathsf{Enc}(1^\lambda, x, m)$	

Figure 6: GGSW's PKE scheme $\mathsf{PKE}[G, \overline{\mathsf{WE}}]$, where G is a length-doubling PRG and $\overline{\mathsf{WE}}$ is a witness encryption scheme for $L_G = \{\, G(s) \,:\, s \in \{0,1\}^* \,\}$

Then $\Pr[\mathrm{AS}^A_{\mathsf{WE_R[P]},L}(\cdot)] = \Pr[\mathrm{IO}^B_{\mathsf{P}}(\cdot)]$ and thus $\mathsf{Adv}^{\mathsf{as}}_{\mathsf{WE_R[P]},L,A}(\cdot) = \mathsf{Adv}^{\mathsf{io}}_{\mathsf{P},B}(\cdot)$. ▮

4 Insufficiency of Soundness Security

GGSW [19] present constructions of several primitives from witness encryption, including PKE, IBE and ABE for all circuits. They claim security of these constructions assuming soundness security of the underlying witness-encryption scheme. We observe here that these claims are wrong. Taking their PRG-based PKE scheme as a representative example, we present a counter-example, namely a witness-encryption scheme satisfying soundness security such that the PKE scheme built from it is insecure. Similar counter-examples can be built for the other applications in GGSW [19]. Briefly, the problem is that a witness encryption scheme could fail to provide any security when $|x|$ is equal to, or related in some specific way to, the security parameter, yet satisfy SS security because the latter requirement holds x fixed and lets λ go to ∞. We show that the gap can be filled, and all the applications of GGSW recovered, by using adaptive soundness in place of soundness security. We'll begin by recalling the well-known notions of PRG and PKE.

Primitives. A pseudorandom generator (PRG) [11, 37] is a PT deterministic algorithm G that takes any string $s \in \{0,1\}^*$ as input and return a string $G(s)$ of length $\ell(|s|)$, where the function $\ell : \mathbb{N} \to \mathbb{N}$ is call the *expansion factor* of G. We say that G is secure if $\mathsf{Adv}^{\mathsf{prg}}_{A,G}(\lambda) = 2\Pr[\mathrm{PRG}^G_A(\lambda)] - 1$ is negligible, for every PT adversary A, where game PRG is defined in Fig. 5.

A public-key encryption (PKE) scheme PKE defines PT algorithms $\mathsf{PKE.Kg}$, $\mathsf{PKE.Enc}$, $\mathsf{PKE.Dec}$, the last deterministic. Algorithm $\mathsf{PKE.Kg}$ takes as input 1^λ and outputs a public encryption key pk and a secret decryption key sk. Algorithm $\mathsf{PKE.Enc}$ takes as input pk and a message $m \in \{0,1\}^*$, and outputs a ciphertext c. Algorithm $\mathsf{PKE.Dec}(sk, c)8$ outputs $m \in \{0,1\}^* \cup \{\bot\}$. Scheme PKE is INDCPA-secure [27, 6] if $\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathsf{PKE},A}(\cdot) = 2\Pr[\mathrm{INDCPA}^A_{\mathsf{PKE}}(\cdot)] - 1$ is negligible for every PT adversary A, where game INDCPA is defined in Fig. 5.

SS does not suffice for GGSW's PKE scheme. Let G be a PRG that is length doubling, meaning $|G(s)| = 2|s|$ for every $s \in \{0,1\}^*$. Let $L_G = \{\, G(s) \,:\, s \in \{0,1\}^* \,\}$. This language is in **NP**. Let $\overline{\mathsf{WE}} \in \mathsf{SS}[L_G]$ be a $\mathsf{SS}[L_G]$-secure WE scheme. The PKE scheme $\mathsf{PKE}[G, \overline{\mathsf{WE}}]$ of

GGSW is shown in Fig. 6. We claim that SS[L_G]-security of $\overline{\mathsf{WE}}$ is insufficient for PKE to be INDCPA-secure. We show this by counter-example, meaning we give an example of a particular WE scheme $\overline{\mathsf{WE}} \in \mathsf{SS}[L_G]$ such that $\mathsf{PKE}[G, \overline{\mathsf{WE}}]$ is not INDCPA. We assume there exists some $\mathsf{WE} \in \mathsf{SS}[L_G]$, else the question is moot. Let $f(\lambda) = 2\lambda$ for every $\lambda \in \mathbb{N}$. Now let $\overline{\mathsf{WE}} = \mathsf{WE}_f$ be the WE scheme of Fig. 3 obtained from WE and f. Lemma 3.1 tells us that $\mathsf{WE}_f \in \mathsf{SS}[L_G]$. Now we claim that $\mathsf{PKE}[G, \mathsf{WE}_f]$ is not INDCPA. The reason is that when $\mathsf{PKE.Enc}(pk, m)$ runs $\mathsf{WE}_f.\mathsf{Enc}(1^\lambda, x, m)$, we have $|x| = 2\lambda = f(\lambda)$. By definition of $\mathsf{WE}_f.\mathsf{Enc}$, the latter returns $(0, m)$ as the ciphertext, effectively sending the message in the clear.

AS security suffices for GGSW's PKE. We now show that the gap can be filled using AS. That is, we prove that if G is a secure PRG and $\overline{\mathsf{WE}}$ is AS[L_G]-secure, then $\mathsf{PKE}[G, \mathsf{WE}]$ is INDCPA-secure:

Theorem 4.1 Let $G : \{0,1\}^* \to \{0,1\}^*$ be a length-doubling PRG. Let $L_G = \{G(s) : s \in \{0,1\}^*\}$. If G is a secure PRG and $\overline{\mathsf{WE}} \in \mathsf{AS}[L_G]$ then $\mathsf{PKE}[G, \overline{\mathsf{WE}}]$ is INDCPA-secure.

The proof is in Appendix D. It follows the template of the proof of GGSW [19]. First one uses the PRG security of G to move to a game where x is random. Since G is length doubling, such an x is not in L_G with high probability. At this point GGSW [19] (incorrectly) claim that the result follows from the SS[L_G]-security of $\overline{\mathsf{WE}}$. We instead use the AS[L_G]-security of $\overline{\mathsf{WE}}$, providing a reduction with an explicit construction of an AS adversary.

Further counter-examples. Actually, GGSW don't use a generic scheme $\overline{\mathsf{WE}} \in \mathsf{SS}[L_G]$ for their PKE scheme. They start with a scheme $\mathsf{WE} \in \mathsf{SS}[L]$ for an **NP**-complete language L, transform it to $\overline{\mathsf{WE}} \in \mathsf{SS}[L_G]$ via the transform in Appendix B, and then define their scheme as $\mathsf{PKE}[G, \overline{\mathsf{WE}}]$. Their proof, however, does not attempt to rely on anything more than the fact that $\overline{\mathsf{WE}} \in \mathsf{SS}[L_G]$. For clarity and simplicity we have accordingly looked at the PKE scheme obtained directly from an arbitrary $\overline{\mathsf{WE}} \in \mathsf{SS}[L_G]$. However, one might ask whether the specific way in which GGSW obtain $\overline{\mathsf{WE}}$ could result in $\mathsf{PKE}[G, \overline{\mathsf{WE}}]$ being secure assuming $\mathsf{WE} \in \mathsf{SS}[L]$. The answer is no. In Appendix C, we show how to extend our counter-example to the actual scheme, meaning that we provide $\overline{\mathsf{WE}} \in \mathsf{SS}[L_G]$, obtained from $\mathsf{WE} \in \mathsf{SS}[L]$ for an **NP**-complete language L via the transform in Fig. 10 such that $\mathsf{PKE}[G, \overline{\mathsf{WE}}]$ fails to be INDCPA-secure.

To obtain similar counter-examples showing the inadequacy of SS for the other applications of GGSW (namely IBE and ABE for all circuits), one can follow the template of our PKE attack, by choosing a lower bound $f(\lambda)$ for the length of the string $x = X(\lambda)$ given to the witness encryption. Since $X(\lambda)$ is generated from some cryptographic primitive π (for example, in IBE, π is a unique signature scheme), the security of π requires that $X(\lambda)$ have super-logarithmic length. Hence there is a constant $C > 0$ such that $|X(\lambda)| \geq C \lg(\lambda)$ for all $\lambda \in \mathbb{N}$, and therefore we can let $f(\lambda) = \lfloor C \lg(\lambda) \rfloor$.

5 Asymmetric Password-based Encryption

In this Section we introduce and define the new primitive of asymmetric password-based encryption (A-PBE). We then provide a non-invasive, WE-based A-PBE scheme we call APBE2, with two security analyses. First we prove security of APBE2 under AS-security of the WE scheme. Then under XS-security of the WE scheme we provide another proof that shows the scheme to admit better "stretch," leading to better security for some real password distributions. In Appendix E we provide a simple and fast, but invasive, A-PBE scheme, called APBE1. Our model and definitions

$\underline{\text{GAME APBE}_{\mathsf{F}}^{A}(\lambda)}$	$\underline{\text{GAME RIP}_{\mathsf{H}}^{A}(\lambda)}$
$\mathbf{pw} \leftarrow\!\!\$\ A_1(1^\lambda)$; $b \leftarrow\!\!\$\ \{0,1\}$	$\mathbf{pw} \leftarrow\!\!\$\ A_1(1^\lambda)$; $b \leftarrow\!\!\$\ \{0,1\}$
For $i = 1$ to $\lvert\mathbf{pw}\rvert$ do	For $i = 1$ to $\lvert\mathbf{pw}\rvert$ do
$\quad \mathbf{sa}[i] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{F.sl}(\lambda)}$	$\quad \mathbf{sa}[i] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.kl}(\lambda)}$
$\quad \mathbf{hpw}[i] \leftarrow \mathsf{F.Ph}(1^\lambda, \mathbf{sa}[i], \mathbf{pw}[i])$	$\quad \mathbf{hpw}[i] \leftarrow \mathsf{H}(1^\lambda, \mathbf{sa}[i], \mathbf{pw}[i])$
$b' \leftarrow\!\!\$\ A_2^{\mathrm{LR}}(1^\lambda, \mathbf{sa}, \mathbf{hpw})$; Return $(b = b')$	$\quad$ If $b = 0$ then $\mathbf{hpw}[i] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.ol}(\lambda)}$
$\underline{\mathrm{LR}(m_0, m_1, i)}$	$b' \leftarrow\!\!\$\ A_2(1^\lambda, \mathbf{sa}, \mathbf{hpw})$; Return $(b = b')$
$c \leftarrow\!\!\$\ \mathsf{F.Enc}(1^\lambda, \mathbf{hpw}[i], \mathbf{sa}[i], m_b)$; Return c	

Figure 7: **Left:** Game APBE defining security of an A-PBE scheme F. **Right:** Game RIP defining RIP security for a hash family H.

are of interest beyond our schemes because they capture PKE in the real-world setting where secret keys are based on passwords and may thus be related.

A-PBE syntax and security. An *asymmetric password-based encryption* (A-PBE) scheme F specifies PT algorithms $\mathsf{F.Ph}, \mathsf{F.Enc}$, $\mathsf{F.Dec}$, the first and the last deterministic. It also specifies a password-length function $\mathsf{F.pl} : \mathbb{N} \to \mathbb{N}$, a salt-length function $\mathsf{F.sl} : \mathbb{N} \to \mathbb{N}$, and a hash-length function $\mathsf{F.hl} : \mathbb{N} \to \mathbb{N}$. Algorithm $\mathsf{F.Ph}$ takes as input the unary representation 1^λ of security parameter λ, a salt $sa \in \{0,1\}^{\mathsf{F.sl}(\lambda)}$, and a password $pw \in \{0,1\}^{\mathsf{F.pl}(\lambda)}$, and returns a hashed password $hpw = \mathsf{F.Ph}(1^\lambda, sa, pw) \in \{0,1\}^{\mathsf{F.hl}(\lambda)}$. Algorithm $\mathsf{F.Enc}$ takes as input $1^\lambda, hpw, sa$ and a message $m \in \{0,1\}^*$, and outputs a ciphertext c. Finally, given (pw, c), algorithm $\mathsf{F.Dec}$ returns $m \in \{0,1\}^* \cup \{\perp\}$. We require that

$$\mathsf{F.Dec}\big(pw, \mathsf{F.Enc}(1^\lambda, \mathsf{F.Ph}(1^\lambda, sa, pw), sa, m)\big) = m$$

for every $m \in \{0,1\}^*, \lambda \in \mathbb{N}, sa \in \{0,1\}^{\mathsf{F.sl}(\lambda)}$, and $pw \in \{0,1\}^{\mathsf{F.pl}(\lambda)}$.

An adversary A is a pair of PT algorithms (A_1, A_2). Adversary $A_1(1^\lambda)$ generates a vector of passwords $\mathbf{pw}$, each entry a $\mathsf{F.pl}(\lambda)$-bit string. It is required that A is unpredictable as defined in Section 2. Note that passwords —entries of the vector $\mathbf{pw}$— may be correlated, even though each individually is unpredictable, to capture the fact that individual users often pick related passwords for their different accounts. We say that A-PBE scheme F is secure if $\mathsf{Adv}^{\mathsf{apbe}}_{\mathsf{F},A}(\cdot) = 2\Pr[\mathrm{APBE}^{A}_{\mathsf{F}}(\cdot)] - 1$ is negligible for every PT unpredictable adversary A, where game $\mathrm{APBE}^{A}_{\mathsf{F}}(\lambda)$ is defined in Fig. 7. In this game, $A_1(1^\lambda)$ first generates its vector $\mathbf{pw}$ of passwords. The game picks a challenge bit $b \leftarrow\!\!\$\ \{0,1\}$ and a vector of random salts $\mathbf{sa}$. Adversary A_2 is given $\mathbf{sa}$ and the vector $\mathbf{hpw}$ of hashed passwords. It can then query its oracle LR with equal-length, distinct messages m_0, m_1, and an index i, to get $\mathsf{F.Enc}(1^\lambda, \mathbf{hpw}[i], \mathbf{sa}[i], m_b)$. Finally A_2 outputs a prediction b' for b. The game returns true if the prediction is correct, meaning $b = b'$, and false otherwise.

Achieving A-PBE. If we have the luxury of prescribing our own password hashing function PH then we can provide a fast and simple A-PBE scheme, that we call APBE1, based on any PKE scheme. See Appendix E. However, this solution is invasive, asking for the deployment of a new PH, which may not be possible due to existing legacy passwords and password-hashing functions. We thus ask if it is possible to design a secure A-PBE scheme that is non-invasive. This means we take $\mathsf{F.Ph}$ as given and aim to achieve security by making reasonable assumptions about its security without prescribing its design, assumptions that in particular are met by the $\mathsf{F.Ph}$ function of PKCS#5 or other standards. This turns out to be more challenging. We now provide the APBE2

$\mathsf{F}[\mathsf{H},\mathsf{WE}].\mathsf{Ph}(1^\lambda, sa, pw)$	$\mathsf{F}[\mathsf{H},\mathsf{WE}].\mathsf{Enc}(1^\lambda, hpw, sa, m)$	$\mathsf{F}[\mathsf{H},\mathsf{WE}].\mathsf{Dec}(pw, c)$
$hpw \leftarrow \mathsf{H}(1^\lambda, sa, pw)$	$x \leftarrow (1^\lambda, sa, hpw)$; $c \leftarrow\!\!\$\; \mathsf{WE}(1^\lambda, x, m)$	$m \leftarrow \mathsf{WE.Dec}(pw, c)$
Return hpw	Return c	Return m

Figure 8: **A-PBE scheme $\mathsf{F} = \mathsf{APBE2}[\mathsf{H},\mathsf{WE}]$ associated to hash family H and witness encryption scheme WE for L_{H}.**

---

scheme that accomplishes this using WE.

Non-invasive A-PBE. We view ourselves as given a function family H with key, input and output length functions $\mathsf{H.kl}, \mathsf{H.il}, \mathsf{H.ol}$. Our goal is to design an A-PBE scheme F such that $\mathsf{F.Ph}$ is H. In particular, we could let H be the password hashing function family from PKCS#5 [33] or bcrypt [34], thereby obtaining A-PBE without change in the existing hashed passwords. We begin by reviewing the security assumption on H.

Related-input pseudorandomness. Let H be a function family. This means that H is a deterministic, PT function taking 1^λ, a key $k \in \{0,1\}^{\mathsf{H.kl}(\lambda)}$ and an input $x \in \{0,1\}^{\mathsf{H.il}(\lambda)}$ to return $\mathsf{H}(1^\lambda, k, x) \in \{0,1\}^{\mathsf{H.ol}(\lambda)}$. Here $\mathsf{H.kl}, \mathsf{H.il}, \mathsf{H.ol}\colon \mathbb{N} \to \mathbb{N}$ are the key, input and output lengths associated to H, respectively. We say that H is related-input pseudorandom (RIP) if $\mathsf{Adv}^{\mathsf{rip}}_{\mathsf{H},A}(\cdot) = 2\Pr[\mathrm{RIP}^A_{\mathsf{H}}(\cdot)] - 1$ is negligible for every PT unpredictable adversary $A = (A_1, A_2)$, where game $\mathrm{RIP}^A_{\mathsf{H}}$ is shown in Fig. 7. Informally, this means that the hashed passwords should be indistinguishable from random strings, even in the presence of the salts. We note that this is exactly the property needed for classical S-PBE (symmetric PBE) to be secure, for it uses the hashed password as the symmetric key. Thus, the assumption can be viewed as already made and existing, even if implicitly, in current usage of passwords for S-PBE. We note that RIP security of H is implied by UCE security of H relative to statistically unpredictable sources [7].

The APBE2 scheme. Let

$$L_{\mathsf{H}} = \{\, (1^\lambda, sa, \mathsf{H}(1^\lambda, sa, pw)) \;:\; \lambda \in \mathbb{N},\ sa \in \{0,1\}^{\mathsf{H.kl}(\lambda)},\ pw \in \{0,1\}^{\mathsf{H.il}(\lambda)} \,\} \;.$$

This language is in **NP**. Let WE be a witness encryption scheme for L_{H}. We associate to H and WE the A-PBE scheme $\mathsf{F} = \mathsf{APBE2}[\mathsf{H},\mathsf{WE}]$ specified in Fig. 8. We let $\mathsf{F.pl} = \mathsf{H.il}$, $\mathsf{F.sl} = \mathsf{H.kl}$ and $\mathsf{F.hl} = \mathsf{H.ol}$. The construction lets the salt play the role of the key for H, the password being the input and the hashed password the output.

Security of APBE2 under AS. Theorem 5.1 below says that if H is RIP and WE is $\mathrm{AS}[L_{\mathsf{H}}]$-secure then $\mathsf{APBE2}[\mathsf{H},\mathsf{WE}]$ is a secure A-PBE scheme. The proof is in Appendix F.

Theorem 5.1 Let H be a function family such that $2^{\mathsf{H.il}(\cdot)-\mathsf{H.ol}(\cdot)}$ is a negligible function. If H is RIP and $\mathsf{WE} \in \mathsf{AS}[L_{\mathsf{H}}]$ then $\mathsf{F} = \mathsf{APBE2}[\mathsf{H},\mathsf{WE}]$ is a secure A-PBE scheme.

The key feature of this result is that it is non-invasive, meaning it puts conditions on the hash family H that suffice for security rather than mandating any particular design of H. Practical and standardized key-derivation functions may be assumed to satisfy concrete versions of these asymptotic conditions.

Arbitrary stretch. Define the stretch $\mathsf{H.s}(\cdot) = \mathsf{H.ol}(\cdot) - \mathsf{H.il}(\cdot)$ of password hashing function H as the difference between its output length and its input length. Theorem 5.1 requires that $2^{-\mathsf{H.s}(\cdot)}$ is negligible, meaning the output length of the hash must be somewhat longer than the input length. This captures situations in which passwords are, say 12-character ASCII strings (input length is

Game $\text{XS}^{A,E}_{\mathsf{WE},\mathsf{R}}(\lambda)$	Game $\text{ROW}^{A}_{\mathsf{H}}(\lambda)$
$(x, m_0, m_1, \text{St}) \leftarrow\!\!\$\, A(1^\lambda)$; $b \leftarrow\!\!\$\, \{0,1\}$	$\mathbf{pw} \leftarrow\!\!\$\, A_1(1^\lambda)$
$c \leftarrow\!\!\$\, \mathsf{WE.Enc}(1^\lambda, x, m_b)$	For $i = 1$ to $\lvert\mathbf{pw}\rvert$ do
$b' \leftarrow\!\!\$\, A(\text{St}, c)$	$\quad \mathbf{sa}[i] \leftarrow\!\!\$\, \{0,1\}^{\mathsf{H.kl}(\lambda)}$; $\mathbf{hpw}[i] \leftarrow \mathsf{H}(1^\lambda, \mathbf{sa}[i], \mathbf{pw}[i])$
$w \leftarrow\!\!\$\, E(1^\lambda, x, m_0, m_1, \text{St}, c)$	$(w, i) \leftarrow\!\!\$\, A_2(1^\lambda, \mathbf{sa}, \mathbf{hpw})$
Return $((b = b') \wedge \neg\mathsf{R}(x, w))$	Return $(\mathbf{hpw}[i] = \mathsf{H}(1^\lambda, \mathbf{sa}[i], w))$

Figure 9: **Left:** Game XS defining extractable security of witness encryption scheme WE. **Right:** Game ROW defining ROW security of H.

78-bit) and H is iterated SHA1 (output length is 160-bit). However, when passwords are longer, say 24-character, then Theorem 5.1 doesn't apply. This is unsatisfying, because intuitively, longer passwords should offer better security. In this section, we formalize a stronger security requirement for witness encryption called XS that allows us to remove the assumption on the stretch of H.

XS-secure witness encryption. The security requirements for SS and AS are for $x \notin L$, no security requirement being made if $x \in L$. Extractable witness encryption [26] is a requirement for all $x \in \{0,1\}^*$, asking that if the adversary violates privacy of encryption under x then one can extract a witness for the membership of $x \in L$. Intuitively, the only way to violate privacy is to know a witness. We provide a formalization of extraction security that we call XS. It strengthens the formalization of GKPVZ [26] in being adaptive, in the vein of AS, but weakens it by not involving auxiliary inputs. The formalizations also differ in other details.

Let R be an **NP**-relation and let $L = \mathcal{L}(\mathsf{R})$. Let WE be a witness encryption scheme for L. We say that WE is XS$[L]$-secure if for any PT adversary A there is a corresponding PT algorithm E such that $\mathsf{Adv}^{\mathsf{xs}}_{\mathsf{WE},\mathsf{R},A,E}(\lambda) = 2\Pr[\text{XS}^{A,E}_{\mathsf{WE},\mathsf{R}}(\lambda)] - 1$ is negligible, where game $\text{XS}^{A,E}_{\mathsf{WE},\mathsf{R}}$ is defined at the leftpanel of Fig. 9. Let $\mathsf{XS}[L]$ denote the set of correct, XS$[L]$-secure witness encryption schemes for L.

Intuitively, XS$[L]$ security implies AS$[L]$ security for any $L \in \mathbf{NP}$, because in the former notion, if the adversary produces $x \notin L$ then no witness exists, so no extractor E (even a computationally unbounded one) can find one. Proposition 5.2 below formally confirms this. The proof is in Appendix H.

Proposition 5.2 For any **NP**-relation R, it holds that $\mathsf{XS}[\mathcal{L}(\mathsf{R})] \subseteq \mathsf{AS}[\mathcal{L}(\mathsf{R})]$.

Extractable obfuscation (xO), also known as differing-input obfuscation, was defined in [4, 13, 1]. BCP [13] show that it implies extractable witness encryption meeting the definition of GKPVZ [26]. In Appendix I, we give an alternative definition of xO and show that it implies XS$[\mathcal{L}(\mathsf{R})]$-secure witness encryption, for any **NP** relation R. The construction is the same $\mathsf{WE}_{\mathsf{R}}[\mathsf{P}]$ in Section 3, where the obfuscator P is assumed to be xO-secure, instead of just being iO-secure.

Related-input one-wayness. We formalize another hardness assumption, namely related-input one-wayness, on hash function family H. Informally we demand that if the adversary is given the hashed passwords and the salts, it can't compute a preimage of any hashed password. This is exactly the intuitive requirement for password-hashing functions: if passwords are well-chosen to resist dictionary attacks, then no adversary should be able to recover some password from the hashed ones. It's a variant of the notion of one-wayness under correlated products of [35]. Formally, we say that H is related-input one-way (ROW) if $\mathsf{Adv}^{\mathsf{row}}_{\mathsf{H},A}(\lambda) = \Pr[\text{ROW}^{A}_{\mathsf{H}}(\lambda)]$ is negligible for all PT unpredictable adversary $A = (A_1, A_2)$, where game $\text{ROW}^{A}_{\mathsf{H}}$ is shown at the right panel of Fig. 9.

Security of APBE2 under XS. The following establishes the security of $\mathsf{F} = \mathsf{APBE2}[\mathsf{H}, \mathsf{WE}]$

without any restrictions or assumptions on the stretch of H. See Appendix G for the proof.

Theorem 5.3 If H is ROW and $\mathsf{WE} \in \mathsf{XS}[L_{\mathsf{H}}]$ then $\mathsf{F} = \mathsf{APBE2}[\mathsf{H}, \mathsf{WE}]$ is a secure A-PBE scheme.

Acknowledgments

We thank Krysztof Pietrzak and Georg Fuchsbauer for discussions about witness encryption.

References

[1] P. Ananth, D. Boneh, S. Garg, A. Sahai, and M. Zhandry. Differing-inputs obfuscation and applications. Cryptology ePrint Archive, Report 2013/689, 2013. `http://eprint.iacr.org/2013/689`. 6, 16, 30

[2] P. V. Ananth, D. Gupta, Y. Ishai, and A. Sahai. Optimizing obfuscation: Avoiding barrington's theorem. In G.-J. Ahn, M. Yung, and N. Li, editors, *ACM CCS 14*, pages 646–658. ACM Press, Nov. 2014. 5

[3] B. Barak, S. Garg, Y. T. Kalai, O. Paneth, and A. Sahai. Protecting obfuscation against algebraic attacks. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 221–238. Springer, May 2014. 5

[4] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. In J. Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 1–18. Springer, Aug. 2001. 6, 11, 16, 30

[5] M. Bellare, R. Canetti, and H. Krawczyk. Keying hash functions for message authentication. In N. Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 1–15. Springer, Aug. 1996. 4

[6] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In H. Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 26–45. Springer, Aug. 1998. 12

[7] M. Bellare, V. T. Hoang, and S. Keelveedhi. Instantiating random oracles via UCEs. Cryptology ePrint Archive, Report 2013/424, 2013. Preliminary version appeared at *CRYPTO 2013*, pages 398–415, 2013. 15

[8] M. Bellare, S. Meiklejohn, and S. Thomson. Key-versatile signatures and applications: RKA, KDM and joint enc/sig. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 496–513. Springer, May 2014. 7

[9] M. Bellare, T. Ristenpart, and S. Tessaro. Multi-instance security and its application to password-based cryptography. In R. Safavi-Naini and R. Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 312–329. Springer, Aug. 2012. 3, 5

[10] M. Bellare and P. Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In S. Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 409–426. Springer, May / June 2006. 8, 24, 27

[11] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudorandom bits. *SIAM Journal on Computing*, 13(4):850–864, 1984. 12

[12] D. Boneh and M. Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 480–499. Springer, Aug. 2014. 19, 20

[13] E. Boyle, K.-M. Chung, and R. Pass. Extractable obfuscation and applications. In Y. Lindell, editor, *Theory of Cryptography*, volume 8349 of *LNCS*, pages 52–73. Springer, 2014. 6, 16, 30

[14] E. Boyle and R. Pass. Limits of extractability assumptions with distributional auxiliary input. Cryptology ePrint Archive, Report 2013/703, 2013. `http://eprint.iacr.org/2013/703`. 6

[15] Z. Brakerski and G. N. Rothblum. Virtual black-box obfuscation for all circuits via generic graded encoding. In Y. Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 1–25. Springer, Feb. 2014. 5

[16] M. Chase and A. Lysyanskaya. On signatures of knowledge. In C. Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 78–96. Springer, Aug. 2006. 7

[17] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49. IEEE Computer Society Press, Oct. 2013. 5, 7, 11, 19, 21

[18] S. Garg, C. Gentry, S. Halevi, and D. Wichs. On the implausibility of differing-inputs obfuscation and extractable witness encryption with auxiliary input. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 518–535. Springer, Aug. 2014. 6

[19] S. Garg, C. Gentry, A. Sahai, and B. Waters. Witness encryption and its applications. In D. Boneh, T. Roughgarden, and J. Feigenbaum, editors, *45th ACM STOC*, pages 467–476. ACM Press, June 2013. 4, 5, 6, 7, 8, 11, 12, 13, 19, 20, 22

[20] S. Garg, C. Gentry, A. Sahai, and B. Waters. Witness encryption and its applications. Cryptology ePrint Archive, Report 2013/258, version 20130508:202916, May 8, 2013. 5, 6, 7, 19

[21] S. Garg, C. Gentry, A. Sahai, and B. Waters. Witness encryption and its applications. Cryptology ePrint Archive, Report 2013/258, version 20140211:224937, February 11, 2014. 7, 19, 20

[22] S. Garg, C. Gentry, A. Sahai, and B. Waters. Witness encryption and its applications. Cryptology ePrint Archive, Report 2013/258, version 20140418:025904, April 18, 2014. 7, 19, 20

[23] C. Gentry, A. Lewko, A. Sahai, and B. Waters. Indistinguishability obfuscation from the multilinear subgroup elimination assumption. Cryptology ePrint Archive, Report 2014/309, 2014. `http://eprint.iacr.org/2014/309`. 5

[24] C. Gentry, A. B. Lewko, and B. Waters. Witness encryption from instance independent assumptions. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 426–443. Springer, Aug. 2014. 7, 21

[25] O. Goldreich. *Foundations of Cryptography: Basic Applications*, volume 2. Cambridge University Press, Cambridge, UK, 2004. 23

[26] S. Goldwasser, Y. T. Kalai, R. A. Popa, V. Vaikuntanathan, and N. Zeldovich. How to run turing machines on encrypted data. In R. Canetti and J. A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 536–553. Springer, Aug. 2013. 6, 16

[27] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984. 12

[28] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989. 9

[29] V. Goyal, A. O'Neill, and V. Rao. Correlated-input secure hash functions. In Y. Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 182–200. Springer, Mar. 2011. 4

[30] S. Hohenberger, A. Sahai, and B. Waters. Replacing a random oracle: Full domain hash from indistinguishability obfuscation. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 201–220. Springer, May 2014. 19, 20

[31] I. Komargodski, M. Naor, and E. Yogev. Secret-sharing for np. In *Advances in Cryptology–Asiacrypt 2014*, pages 254–273. Springer, 2014. 7, 20, 22

[32] N. Nisan and D. Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–52, 1996. 4

[33] PKCS #5: Password-based cryptography standard (RFC 2898). RSA Data Security, Inc., Sept. 2000. Version 2.0. 3, 4, 5, 15

[34] N. Provos and D. Mazières. A future-adaptable password scheme. In *USENIX Annual Technical Conference, FREENIX Track*, pages 81–91, 1999. 3, 4, 15

[35] A. Rosen and G. Segev. Chosen-ciphertext security via correlated products. In O. Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 419–436. Springer, Mar. 2009. 16

[36] A. Sahai and B. Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In D. B. Shmoys, editor, *46th ACM STOC*, pages 475–484. ACM Press, May / June 2014. 19, 20

[37] A. C.-C. Yao. Theory and applications of trapdoor functions (extended abstract). In *23rd FOCS*, pages 80–91. IEEE Computer Society Press, Nov. 1982. 12

A Further versions of SS

A good definition for WE security should have two properties: (1) *Usability*, meaning it should suffice to prove security of applications, and (2) *Achievability*, meaning it should be provably achieved by the natural constructs, which in this case means the iO-based one of GGHRSW [17]. Our AS definition has both properties. We have shown that SS [19, 20] lacks (1).

After seeing a prior version of our work, GGSW updated the ePrint version of their paper [21]. Here they acknowledge the gaps we find. They then propose their own modification of SS, that we call SS2, in an attempt to fill the gaps. This was unnecessary because AS had already been put forth and shown to fill the gap, but GGSW appeared to want a definition in their quantifier-based style rather than our game-based style. They viewed the problem in SS as arising from the "order of quantification" and attempted to address it by changing this order. SS2 quantified the negligible function first, making it universal. We explain below that SS2 is unachievable, meaning no WE scheme can be SS2 secure. (More precisely our result is that SS2-secure WE is unachievable for any **NP**-complete language unless the polynomial-time hierarchy collapses. Our proof uses the fact that statistically-secure WE is not achievable [19].) We pointed this out to GGSW in a personal communication. They acknowledged this and further updated their definition to one we call SS3 [21], which used another order of quantification. Below we show that SS3 remains limited in terms of achievability. This is because it does not seem possible to show that the iO-based WE construction of GGHRSW [17] meets it under the definition of iO-security that is commonly used in other applications of iO [36, 30, 12] and that we have shown suffices for AS-secure WE.

The updated GGSW papers [21, 22] characterize the gap we find as having to do with the "order of quantifiers" in the SS definition, and their fixes attempt to change quantifier order. However, the issue is not quantifier order but, more subtly, the relation between x and λ. More broadly, game-based definitions are a better fit in this domain than quantifier-based ones. This is because applications one wants to achieve with WE, as well as primitives one wants to use to achieve WE, are both themselves underlain by game-based definitions. Reductions are thus facilitated, and less error-prone, with a game-based WE definition. A quantifier-based one leads to mismatches. In particular, under certain quantifier orders, one gets definitions like SS that do not provide usability, and when one changes the order, one gets definitions like SS3 that are too strong and challenge achievability. Intuitively, the latter is because the quantification ends up demanding security even on inputs that no adversary could ever find. This does not mean a viable quantifier-based definition is impossible. Indeed, below, we suggest SS4, a quantifier-based definition of WE that recovers achievability under weak iO in the non-uniform case. But the game-based AS is simpler and more user friendly, and does not require non-uniformity to be achieved under weak iO.

Below we also consider a recent definition of soundness security of KNY [31]. We call it SS5. It is similar to SS2 and consequently also unachievable.

The above indicates that the problems we find with SS, and the fix we deliver with AS, are not trivial. Certainly it is easy, once the problem has been pointed out, to propose alternatives, but our work remains important in having pointed out the need for alternatives and in guiding the choice of, and verifying, these alternatives.

We believe that WE is an important and useful notion and that our work helps advance its cause via precise definitions that satisfy the usability and achievability conditions above. We believe it is important for our field that work like this is published, and that such work is not damaging for the GGSW authors but rather advances the primitive they proposed.

SS2. WE scheme WE is SS2[L]-secure according to [21] if there exists a negligible function $\nu : \mathbb{N} \to \mathbb{N}$ such that for any PT adversary A, any $x \in \{0,1\}^* \backslash L$, any equal-length $m_0, m_1 \in \{0,1\}^*$, and any $\lambda \in \mathbb{N}$ we have

$$\begin{aligned} & \mathsf{Adv}^{\mathsf{ss2}}_{\mathsf{WE},L,x,m_0,m_1,A}(\lambda) \\ = \ & \Pr[A(\mathsf{WE.Enc}(1^\lambda, x, m_1)) = 1] - \Pr[A(\mathsf{WE.Enc}(1^\lambda, x, m_0)) = 1] < \nu(\lambda) \ . \end{aligned}$$

We claim this notion is unachievable, meaning, *no* WE scheme is SS2[L]-secure. The reason is that ν is universal and in particular not allowed to depend on the adversary. More formally let L be an **NP**-complete language. Let WE by any WE scheme and let ν be any negligible function. We show that if the polynomial-time hierarchy does not collapse then there is a PT adversary A as well as $x \in \{0,1\}^* \setminus L$, equal-length m_0, m_1 and $\lambda \in \mathbb{N}$ such that $\mathsf{Adv}^{\mathsf{ss2}}_{\mathsf{WE},L,x,m_0,m_1,A}(\lambda) \geq \nu(\lambda)$. This shows that WE is not SS2[L]-secure.

For probability distribution functions $\mu, \mu' : D \to [0,1]$, let

$$\|\mu - \mu'\| = \frac{1}{2} \sum_{x \in D} |\mu(x) - \mu'(x)|$$

be the statistical distance between μ and μ'. For any λ, x, m let $\mu_{\lambda,x,m}$ be the distribution of $\mathsf{WE.Enc}(1^\lambda, x, m)$. Results from GGSW [19] imply that, unless the polynomial hierarchy collapses, there exists a string $x' \in \{0,1\}^* \backslash L$, equal-length messages m'_0, m'_1 and a constant $\lambda_0 \in \mathbb{N}$ such that $\|\mu_{\lambda_0,x',m'_1} - \mu_{\lambda_0,x',m'_0}\| \geq \nu(\lambda_0)$. Consider the following adversary A. On input a ciphertext c, if c is not in the domain of μ_{λ_0,x',m'_1} then A outputs a random guess. Otherwise, A outputs 1 if $\mu_{\lambda_0,x',m'_1}(c) > \mu_{\lambda_0,x',m'_0}(c)$, and outputs 0 otherwise. Note that the test as to whether c is in the domain of μ_{λ_0,x',m'_1} only takes polynomial time because λ_0, x', m'_1 are fixed, and all computations related to them are constant time, and similarly for computations of $\mu_{\lambda_0,x',m'_0}(\cdot)$. Thus A runs in polynomial time. But $\mathsf{Adv}^{\mathsf{ss2}}_{\mathsf{WE},L,x',m'_0,m'_1,A}(\lambda_0) = \|\mu_{\lambda_0,x',m'_1} - \mu_{\lambda_0,x',m'_0}\| \geq \nu(\lambda_0)$.

SS3. A WE scheme WE is SS3[L]-secure [22] if for any PT adversary A, there exists a negligible function $\nu : \mathbb{N} \to \mathbb{N}$ such that for any $x \in \{0,1\}^* \backslash L$ and any $\lambda \in \mathbb{N}$,

$$\begin{aligned} & \mathsf{Adv}^{\mathsf{ss3}}_{\mathsf{WE},L,x,A}(\lambda) \\ = \ & \Pr[A(\mathsf{WE.Enc}(1^\lambda, x, 1)) = 1] - \Pr[A(\mathsf{WE.Enc}(1^\lambda, x, 0)) = 1] < \nu(\lambda) \ . \end{aligned}$$

A first nit is that this considers only encryption of a 1-bit message but for applications one has to encrypt many bits, and it is not stated how security is defined in this case. More importantly, however, SS3 has limitations with regard to achievability. Specifically, it seems unlikely one can show that iO implies SS3[L]-secure WE via the natural GGHRSW construction that worked for both SS and AS and under the definition of iO that is used for other applications [36, 30, 12] and we have shown suffices for AS-secure WE. We now explain, referring to our formulation of the definition in Section 3. Let L be an **NP** language. In Section 3 we recalled the GGHRSW construction $\mathsf{WE} = \mathsf{WE}_{\mathsf{R}}[\mathsf{P}]$ of a WE scheme from an indistinguishability obfuscator P. Now assume we are given an arbitrary PT adversary A attacking the SS3[L]-security of WE. To prove security, we need to build an adversary B attacking the iO-security of P. Adversary B, given 1^λ, needs to efficiently find and output circuits of the form $R_{x,m}$ that we defined in Section 3, where x intuitively is an input where the WE security "breaks." But how is B to find such an x efficiently? There seems to be no way. Even if we allow B to be non-uniform, its advice string has length polynomial in λ, and thus it can't tell what is the "best" x because the set $\{0,1\}^*\backslash L$ is infinite. In the case of AS, this was not a problem because A handed back an x on which it succeeded. Also for the original SS, it is not a problem because the entire claim pertains to only one, fixed x that can be assumed known to B. An approach we might consider for SS3 is the following. Given any string $x \notin L$, we can build an adversary B_x such that $\mathsf{Adv}^{\mathsf{ss3}}_{\mathsf{WE},L,x,A}(\lambda) \leq \mathsf{Adv}^{\mathsf{io}}_{\mathsf{P},B_x}(\lambda)$ for all $\lambda \in \mathbb{N}$. Now the assumed iO security gives us a negligible function ν_x such that $\mathsf{Adv}^{\mathsf{io}}_{\mathsf{P},B_x}(\cdot) < \nu_x(\cdot)$. But the SS3 notion wants a single negligible function ν that is independent of x. It's unclear how to get ν from the set $\{\nu_x : x \notin L\}$ since the latter set is infinite. One natural idea is to set $\nu(\lambda) = \sup_{x \notin L}\{\nu_x(\lambda)\}$. But this doesn't work. For example, consider $\nu_x(\lambda) = 1$ if $\lambda < |x|$, and $\nu_x(\lambda) = 0$ otherwise. For any fixed x, the function ν_x is negligible, but $\nu(\lambda) = 1$ for every $\lambda \in \mathbb{N}$, meaning ν is not negligible. So one appears to need to construct an iO adversary B independent of x, but it is unclear how to do that.

We note the proof does seem possible under some stronger notions of iO from [17]. However, iO is a strong assumption no matter what and it is desirable that applications use as weak a form of it as possible. Also in further work subsequent to ours, GLW [24] claim to achieve SS3 via a direct construction. However, this requires sub-exponential hardness assumptions. (They call it complexity leveraging.) Beyond this, trying to achieve SS3 is an unnecessary route to follow, since AS already provides the properties we want, namely it suffices for applications and is achieved even under weak iO.

SS4. As we have seen, the game-based AS definition fulfills the usability and achievability conditions for a good definition. However, GGSW appear to want a quantifier-based definition in the style of SS. But their SS2, SS3 attempts have been inadequate. Here we accordingly suggest a quantifier-based definition that we call SS4 which does satisfy, usability and is less limited than SS3 with regard to achievability, namely weak iO does suffice for it, as long as this is assumed for non-uniform adversaries. In particular it is implied by the non-uniform generalization of AS and thus can be built from weak, non-uniform iO by our results. We explain why it suffices for the PKE application of GGSW.

We say that a WE scheme WE is SS4[L]-secure if for any PT adversary A and any polynomial $\ell : \mathbb{N} \to \mathbb{N}$, there exists a negligible function $\nu : \mathbb{N} \to \mathbb{N}$ such that, for any string $x \in \{0,1\}^*\backslash L$, and any $\lambda \in \mathbb{N}$, if $|x| \leq \ell(\lambda)$ then

$$\begin{aligned} & \mathsf{Adv}^{\mathsf{ss4}}_{\mathsf{WE},L,\ell,x,A}(\lambda) \\ = \;& \Pr[A(1^\lambda, x, \mathsf{WE.Enc}(1^\lambda, x, 1)) = 1] - \Pr[A(1^\lambda, x, \mathsf{WE.Enc}(1^\lambda, x, 0)) = 1] \\ < \;& \nu(\lambda) \ . \end{aligned}$$

We now show this notion is implied by non-uniform AS. Given any SS4 adversary A and any polynomial ℓ, one can build another non-uniform AS adversary B as follows. For each $\lambda \in \mathbb{N}$, let $x_\lambda \in \{0,1\}^*\backslash L$ be a string such that $|x_\lambda| \leq \ell(\lambda)$ and $\mathsf{Adv}^{\mathsf{ss4}}_{\mathsf{WE},L,\ell,x,A}(\lambda) \leq \mathsf{Adv}^{\mathsf{ss4}}_{\mathsf{WE},L,\ell,x_\lambda,A}(\lambda)$ for all $x \in \{0,1\}^*\backslash L$ with $|x| \leq \ell(\lambda)$. Adversary $B(1^\lambda)$ outputs $(x_\lambda, 0, 1, \varepsilon)$, and $B(\mathrm{St}, c)$ runs $A(1^\lambda, x_\lambda, c)$.

Then for any string $x \in \{0,1\}^* \backslash L$, $\lambda \in \mathbb{N}$, if $|x| \le \ell(\lambda)$ then

$$\mathsf{Adv}^{\mathsf{as}}_{\mathsf{WE},B}(\lambda) = \mathsf{Adv}^{\mathsf{ss4}}_{\mathsf{WE},L,\ell,x_\lambda,A}(\lambda) \ge \mathsf{Adv}^{\mathsf{ss4}}_{\mathsf{WE},L,\ell,x,A}(\lambda) \ .$$

We now briefly explain why SS4 is enough for GGSW's PKE scheme, but under message space $\{0,1\}$, because SS4 only allows encrypting a single bit. The INDCPA adversary is assumed to make only a single query $(0,1)$. The proof will follow the template in Appendix D but with a change in constructing WE adversary D from an INDCPA adversary A. Let $\ell(\lambda) = 2\lambda$ for every $\lambda \in \mathbb{N}$. Adversary $D(1^\lambda, x, c)$ runs $A(1^\lambda, pk)$ with $pk = (\lambda, x)$. When the latter makes its query, the former returns c. Finally, D outputs the same guess as A.

For both SS3 and SS4, in the PKE application, to encrypt an n-bit message, one has to make n calls to WE to encrypt n bits individually, exacerbating the inefficiency of the scheme. If one modifies SS3 and SS4 for encrypting equal-length m_0, m_1 of arbitrary length instead of $m_0 = 0$ and $m_1 = 1$, then the PKE still can only encrypt bit-by-bit. The reason is that, an INDCPA adversary A is allowed to choose any equal-length m_0, m_1 but in SS3 and SS4, the WE adversary D has no control of the messages m_0, m_1, and thus one can't construct D from A. This again shows that AS is superior to SS3 and SS4 in terms of usability.

SS5. In a recent paper, KNY [31] define the following variant of SS, which we call SS5. A scheme is SS5$[L]$-secure if for any security parameter λ, any equal-length messages $m_0, m_1 \in \{0,1\}^{\mathrm{poly}(\lambda)}$, any PT adversary A, and any $x \notin L$, we have

$$\begin{aligned} & \mathsf{Adv}^{\mathsf{ss5}}_{\mathsf{WE},L,x,m_0,m_1,A}(\lambda) \\ = \ & \Pr[A(\mathsf{WE.Enc}(1^\lambda, x, m_1) = 1] - \Pr[A(\mathsf{WE.Enc}(1^\lambda, x, m_0)) = 1] < \mathsf{negl}(\lambda) \ . \end{aligned}$$

This definition doesn't specify where to place the (existential) quantifier for the negligible function negl, but the only meaningful position in the context of what is written is to place it prior to the (universal) quantification of the security parameter. (We certainly don't want a different negligible function for every value of λ.) But if so, the function negl is independent of the adversary A. The same argument against SS2 can be used to show that SS5 is unachievable.

SS5 again demonstrates that quantifier-based notions for WE are error-prone. KNY's definition [31] is problematic, although it is subsequent to our work and all of GGSW's revisions.

B WE for any NP language from WE for an NPC language

A Levin reduction from R_2 to R_1 is a triple of PT-computable functions (g, μ, ν) such that (i) $g(x) \in \mathcal{L}(\mathsf{R}_1)$ if and only if $x \in \mathcal{L}(\mathsf{R}_2)$, (ii) If $x \in \mathcal{L}(\mathsf{R}_2)$ and $w \in \mathsf{R}_2(x)$ then $\mu(x,w) \in \mathsf{R}_1(g(x))$, and (iii) If $x \in \mathcal{L}(\mathsf{R}_2)$ and $z \in \mathsf{R}_1(g(x))$ then $\nu(g(x), z) \in \mathsf{R}_2(x)$.

Let $\mathsf{R}_1, \mathsf{R}_2$ be **NP**-relations such that there is a Levin reduction (g, μ, ν) from R_2 to R_1. The transform $\mathrm{Trans}_{g,\mu}$ in Fig. 10 describes how to transform a witness encryption scheme for $\mathcal{L}(\mathsf{R}_1)$ to a witness encryption scheme for $\mathcal{L}(\mathsf{R}_2)$. Claim (1) of Proposition B.1 below is implicit in [19].

Proposition B.1 Let $\mathsf{R}_1, \mathsf{R}_2$ be **NP**-relations such that there is a Levin reduction (g, μ, ν) from R_2 to R_1. Let $\mathrm{Trans}_{g,\mu}$ be the transform specified in Fig. 10 and WE_1 be a witness encryption scheme for $\mathcal{L}(R_1)$. Let $\mathsf{WE}_2 = \mathrm{Trans}_{g,\mu}(\mathsf{WE}_1)$. (1) If $\mathsf{WE}_1 \in \mathsf{SS}[\mathcal{L}(\mathsf{R}_1)]$ then $\mathsf{WE}_2 \in \mathsf{SS}[\mathcal{L}(\mathsf{R}_2)]$, and (2) If $\mathsf{WE}_1 \in \mathsf{AS}[\mathcal{L}(\mathsf{R}_1)]$ then $\mathsf{WE}_2 \in \mathsf{AS}[\mathcal{L}(\mathsf{R}_2)]$.

Proof: For part (1), let A be a PT adversary. Consider arbitrary $x \in \{0,1\}^* \backslash \mathcal{L}(\mathsf{R}_2)$ and $m_0, m_1 \in \{0,1\}^*$ such that $|m_0| = |m_1|$. Note that $g(x) \in \{0,1\}^* \backslash \mathcal{L}(\mathsf{R}_1)$. Then

$$\mathsf{Adv}^{\mathsf{ss}}_{\mathsf{WE}_2,\mathcal{L}(\mathsf{R}_2),x,m_0,m_1,A}(\lambda) = \mathsf{Adv}^{\mathsf{ss}}_{\mathsf{WE}_1,\mathcal{L}(\mathsf{R}_1),g(x),m_0,m_1,A}(\lambda)$$

$\underline{\mathsf{WE}_2.\mathsf{Enc}(1^\lambda, x, m)}$	$\underline{\mathsf{WE}_2.\mathsf{Dec}(c, w)}$
$x' \leftarrow g(x)$; $c' \leftarrow\!\!\$\ \mathsf{WE}_1.\mathsf{Enc}(1^\lambda, x', m)$	$(x, c') \leftarrow c$; $w' \leftarrow \mu(x, w)$
Return (x, c')	$m \leftarrow\!\!\$\ \mathsf{WE}_1.\mathsf{Dec}(c', w')$; Return m

Figure 10: Witness encryption scheme $\mathsf{WE}_2 = \mathrm{Trans}_{g,\mu}(\mathsf{WE}_1)$ for $\mathcal{L}(\mathsf{R}_2)$, with $\mathsf{WE}_2.\mathsf{Msg} = \mathsf{WE}_1.\mathsf{Msg}$, where $\mathsf{R}_1, \mathsf{R}_2$ are **NP**-relations, WE_1 is a witness encryption scheme for $\mathcal{L}(\mathsf{R}_1)$, and (g, μ, ν) is a Levin reduction from R_2 to R_1.

---

for every $\lambda \in \mathbb{N}$, and thus $\mathsf{WE}_2 \in \mathsf{SS}[\mathcal{L}(\mathsf{R}_2)]$.

For part (2), let A be a PT adversary attacking WE_2. Consider the following adversary B attacking WE_1.

$\underline{B(1^\lambda)}$	$\underline{B(\mathrm{St}, c)}$
$(x, m_0, m_1, \mathrm{St}) \leftarrow\!\!\$\ A(1^\lambda)$; $x' \leftarrow g(x)$	$(x, c') \leftarrow c$; $b' \leftarrow\!\!\$\ A(\mathrm{St}, c')$
Return $(x', m_0, m_1, \mathrm{St})$	Return b'

Then $\mathsf{Adv}^{\mathsf{as}}_{\mathsf{WE}_1,\mathcal{L}(\mathsf{R}_1),B}(\lambda) = \mathsf{Adv}^{\mathsf{as}}_{\mathsf{WE}_2,\mathcal{L}(\mathsf{R}_2),A}(\lambda)$ for every $\lambda \in \mathbb{N}$, and thus $\mathsf{WE}_2 \in \mathsf{AS}[\mathcal{L}(\mathsf{R}_2)]$. ▮

C Extending our counter-examples

Recall that in Section 4, we have built a counter-example for scheme $\mathsf{PKE}[G, \overline{\mathsf{WE}}]$ (specified in Fig. 6) where G is a length-doubling PRG and $\overline{\mathsf{WE}}$ is a generic SS-secure witness encryption scheme for $L_G = \{\, G(s) : s \in \{0,1\}^* \,\}$. However, GGSW start with a scheme $\mathsf{WE} \in \mathsf{SS}[L]$ for an **NP**-complete language $L = \mathcal{L}(\mathsf{R})$, transform it to $\overline{\mathsf{WE}} \in \mathsf{SS}[L_G]$ via the transform in Fig. 10 and then define their scheme as $\mathsf{PKE}[G, \overline{\mathsf{WE}}]$. We now extend our counter-example to the actual scheme.

Let R_G be the **NP**-relation of L_G, namely $\mathsf{R}_G(x, w)$ returns $(x = G(w))$. Let (g, μ, ν) be a Levin reduction from L_G to L. In the actual scheme, one obtains $\overline{\mathsf{WE}} \in \mathsf{SS}[L_G]$ via $\mathrm{Trans}_{g,\mu}(\mathsf{WE})$, where WE is a SS$[L]$-secure witness encryption scheme, and $\mathrm{Trans}_{g,\mu}$ is specified in Fig. 10. Since function ν is PT-computable, there are constants $C, d \geq 1$ such that $\mathsf{R}_G.\mathsf{wl}(u) \leq C \cdot |g(u)|^d$, for every $u \in L_G$. Consider arbitrary $\mathsf{WE} \in \mathsf{SS}[L]$ and let $f(\lambda) = \lfloor \frac{\lambda^{1/d}}{C} \rfloor$ for every $\lambda \in \mathbb{N}$. By way of Lemma 3.1, we can modify WE to $\mathsf{WE}_f \in \mathsf{SS}[L]$ (as specified in Fig. 3) that misbehaves, returning the message in the clear when $|x| \geq f(\lambda)$. When we run scheme $\mathsf{PKE}[G, \mathrm{Trans}_{g,\mu}(\mathsf{WE}_f)]$, we always give $\mathsf{WE}_f(1^\lambda, \cdot, m)$ the string $x = g(u)$ for some $u \in L_G \cap \{0,1\}^{2\lambda}$, and thus $|x| \geq f(\mathsf{R}_G.\mathsf{wl}(u)) = f(\lambda)$. Hence $\mathsf{PKE}[G, \mathrm{Trans}_{g,\mu}(\mathsf{WE}_f)]$ always sends messages in the clear.

D Proof of Theorem 4.1

Let A be a PT attacking $\mathsf{PKE}[G, \mathsf{WE}]$. Since one-message INDCPA implies multiple-message INDCPA [25, Theorem 5.2.11], wlog, assume that A makes only a single query. Let ρ denote the coin length of A. Consider the following adversaries B and D:

$\underline{B(1^\lambda, x)}$
$pk \leftarrow (\lambda, x)$; $b \leftarrow\!\!\$\ \{0,1\}$
$b' \leftarrow\!\!\$\ A^{\mathrm{LRSIM}}(1^\lambda, pk)$
If $b = b'$ then return 1 else return 0

$\underline{\mathrm{LRSIM}(m_0, m_1)}$
$c \leftarrow\!\!\$\ \mathsf{WE.Enc}(1^\lambda, x, m_b)$; Return c

$\underline{D(1^\lambda)}$
$x \leftarrow\!\!\$\ \{0,1\}^{2\lambda}$; $pk \leftarrow (\lambda, x)$; $r \leftarrow\!\!\$\ \{0,1\}^{\rho(\lambda)}$; $c \leftarrow \bot$
$A^{\mathrm{LRSIM}}(1^\lambda, pk; r)$; $\mathrm{St} \leftarrow (\lambda, pk, r)$; Return (x, m_0, m_1, t)

$\underline{D(\mathrm{St}, c)}$
$(\lambda, pk, r) \leftarrow \mathrm{St}$; $b' \leftarrow A^{\mathrm{LRSIM}}(1^\lambda, pk; r)$; Return b'

$\underline{\mathrm{LRSIM}(m_0, m_1)}$
Return c

Consider games H_1–H_3 below, in which game H_3 includes the boxed statement but game H_2 does not.

$\underline{\text{GAME } H_1^A(\lambda)}$
$s \leftarrow\!\!\$\ \{0,1\}^\lambda$; $x \leftarrow G(s)$
$b \leftarrow\!\!\$\ \{0,1\}$; $\mathsf{passed} \leftarrow \mathsf{true}$
$pk \leftarrow (\lambda, x)$; $b' \leftarrow\!\!\$\ A^{\mathrm{LR}}(1^\lambda, pk)$
Return $(b = b') \wedge \mathsf{passed}$

$\underline{\mathrm{LR}(m_0, m_1)}$
$c \leftarrow\!\!\$\ \mathsf{WE.Enc}(1^\lambda, x, m_b)$; Return c

$\underline{\text{GAME } H_2^A(\lambda), \boxed{H_3^A(\lambda)}}$
$x \leftarrow\!\!\$\ \{0,1\}^{2\lambda}$; $b \leftarrow\!\!\$\ \{0,1\}$; $\mathsf{passed} \leftarrow \mathsf{true}$
If $x \in L_G$ then $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{\mathsf{passed} \leftarrow \mathsf{false}}$
$pk \leftarrow (\lambda, x)$; $b' \leftarrow\!\!\$\ A^{\mathrm{LR}}(1^\lambda, pk)$
Return $(b = b') \wedge \mathsf{passed}$

$\underline{\mathrm{LR}(m_0, m_1)}$
$c \leftarrow\!\!\$\ \mathsf{WE.Enc}(1^\lambda, x, m_b)$; Return c

On the one hand,

$$\Pr[\,\mathrm{PRG}_G^B(\lambda) \Rightarrow \mathsf{true} \,|\, a = 1\,] = \Pr[H_1^A(\lambda)] \qquad \text{and} \qquad \Pr[\,\mathrm{PRG}_G^B(\lambda) \Rightarrow \mathsf{false} \,|\, a = 0\,] = \Pr[H_2^A(\lambda)]$$

for every $\lambda \in \mathbb{N}$, where a is the challenge bit of game PRG_G^B. On the other hand, games H_2 and H_3 are identical-until-bad, and from the fundamental lemma of game-playing [10],

$$\Pr[H_2^A(\lambda)] - \Pr[H_3^A(\lambda)] \le \Pr[H_3^A(\lambda) \text{ sets } \mathsf{bad}] \le 2^{-\lambda}$$

for every $\lambda \in \mathbb{N}$; the last inequality is due to the fact that $L_G \cap \{0,1\}^{2\lambda} = \{\, G(s) \ :\ s \in \{0,1\}^\lambda \,\}$ contains at most 2^λ elements. Moreover,

$$\Pr[\mathrm{INDCPA}^A_{\mathsf{PKE}[G,\mathsf{WE}]}(\lambda)] = \Pr[H_1^A(\lambda)], \qquad \text{and} \qquad \Pr[\mathrm{AS}^D_{\mathsf{WE},L_G}(\lambda)] = \Pr[H_3^A(\lambda)]$$

for every $\lambda \in \mathbb{N}$. Summing up, $\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathsf{PKE}[G,\mathsf{WE}],A}(\lambda) \le 2\mathsf{Adv}^{\mathsf{prg}}_{G,B}(\lambda) + \mathsf{Adv}^{\mathsf{as}}_{\mathsf{WE},L_G,D}(\lambda) + 2^{1-\lambda}$ for every $\lambda \in \mathbb{N}$, and thus $\mathsf{PKE}[G, \mathsf{WE}]$ is INDCPA-secure.

E The APBE1 scheme

Here we describe a simple and fast, but invasive, A-PBE scheme, derived from a RIP function family H and a PKE scheme PKE.

Results. Let PKE be a PKE scheme and H a function family such that $\mathsf{H.ol}$ is the number of coins used by $\mathsf{PKE.Kg}$. Associate to them the A-PBE scheme $\mathsf{F} = \mathsf{APBE1}[\mathsf{H}, \mathsf{PKE}]$ whose constituent algorithms are shown in Fig. 11. Algorithm $\mathsf{F.Ph}(1^\lambda, sa, pw)$ applies H to $1^\lambda, sa, pw$ to get a string r, uses the latter as coins to deterministically compute $(pk, sk) \leftarrow \mathsf{PKE.Kg}(1^\lambda; r)$, and finally returns $hpw = pk$ as the "hashed password." Algorithm $\mathsf{F.Enc}(1^\lambda, pk, sa, m)$ returns ciphertext $(1^\lambda, sa, \mathsf{PKE.Enc}(pk, m))$. Finally, $\mathsf{F.Dec}(pw, (1^\lambda, sa, y))$ re-applies H to $1^\lambda, sa, pw$ to get r, then re-computes $(pk, sk) \leftarrow \mathsf{PKE.Kg}(1^\lambda; r)$ and returns $m \leftarrow \mathsf{PKE.Dec}(sk, y)$. Theorem E.1 below shows that F is a secure A-PBE scheme.

$\underline{\mathsf{F.Ph}(1^\lambda, sa, pw)}$	$\underline{\mathsf{F.Enc}(1^\lambda, hpw, sa, m)}$	$\underline{\mathsf{F.Dec}(pw, c)}$
$r \leftarrow \mathsf{H}(1^\lambda, sa, pw)$	$y \leftarrow\$\ \mathsf{PKE.Enc}(hpw, m)$	$(1^\lambda, sa, y) \leftarrow c\ ;\ r \leftarrow \mathsf{H}(1^\lambda, sa, pw)$
$(pk, sk) \leftarrow \mathsf{PKE.Kg}(1^\lambda; r)$	$c \leftarrow (1^\lambda, sa, y)$	$(pk, sk) \leftarrow \mathsf{PKE.Kg}(1^\lambda; r)\ ;\ m \leftarrow \mathsf{PKE.Dec}(sk, y)$
Return pk	Return c	Return m

Figure 11: A-PBE scheme $\mathsf{F} = \mathsf{APBE1}[\mathsf{H}, \mathsf{PKE}]$ associated to hash function family H and public-key encryption scheme PKE.

$\underline{\text{GAME } H_1^A(\lambda), \boxed{H_2^A(\lambda)}}$

$b \leftarrow\$\ \{0,1\}\ ;\ \mathbf{pw} \leftarrow\$\ A_1(1^\lambda)$

For $i = 1$ to $|\mathbf{pw}|$ do

$\quad \mathbf{sa}[i] \leftarrow\$\ \{0,1\}^{\mathsf{H.kl}(\lambda)}\ ;\ \mathbf{r}[i] \leftarrow \mathsf{H}(1^\lambda, \mathbf{sa}[i], \mathbf{pw}[i])$

$\quad \boxed{\mathbf{r}[i] \leftarrow\$\ \{0,1\}^{\mathsf{H.ol}(\lambda)}\ ;}\ (\mathbf{pk}[i], \mathbf{sk}[i]) \leftarrow \mathsf{PKE.Kg}(1^\lambda; \mathbf{r}[i])$

$b' \leftarrow\$\ A_2^{\mathrm{LR}}(1^\lambda, \mathbf{sa}, \mathbf{pk})\ ;\ $ Return $(b' = b)$

$\underline{\mathrm{LR}(m_0, m_1, i)}$

$y \leftarrow\$\ \mathsf{PKE.Enc}(\mathbf{pk}[i], m_b)\ ;\ $ Return $(1^\lambda, \mathbf{sa}[i], y)$

Figure 12: Games in the proof of Theorem E.1.

Theorem E.1 Let H be a RIP-secure function family. Let PKE be an INDCPA-secure PKE scheme. Let $\mathsf{F} = \mathsf{APBE1}[\mathsf{H}, \mathsf{PKE}]$ be the A-PBE scheme defined above. Then F is a secure A-PBE scheme.

Proof: Let $A = (A_1, A_2)$ be a PT adversary attacking F. Let p be a polynomial that bounds the number of entries in the password vector that A_1 produces. Consider adversaries $B = (B_1, B_2)$ and D in Fig. 13. Since B_1 is exactly A_1, adversary B is unpredictable. Consider games H_1 and H_2 in Fig. 12, in which game H_2 includes the boxed statement but game H_1 does not. Then $\Pr[\mathrm{APBE}_{\mathsf{F}}^A(\cdot)] = \Pr[H_1^A(\cdot)]$ and $\Pr[\mathrm{INDCPA}_{\mathsf{H}}^D(\cdot)] \geq \frac{1}{p}\Pr[H_2^A(\cdot)]$. On the other hand,

$$\begin{aligned} \Pr[\,\mathrm{RIP}_{\mathsf{H}}^B(\cdot)\,|\,a = 1\,] &= \Pr[H_1^A(\cdot)], \text{ and} \\ \Pr[\,\mathrm{RIP}_{\mathsf{H}}^B(\cdot)\,|\,a = 0\,] &= 1 - \Pr[H_2^A(\cdot)], \end{aligned}$$

where a is the challenge bit of game $\mathrm{RIP}_{\mathsf{H}}^B$. Summing up, $\mathsf{Adv}_{\mathsf{F},A}^{\mathsf{apbe}}(\cdot) \leq \mathsf{Adv}_{\mathsf{H},B}^{\mathsf{rip}}(\cdot) + p \cdot \mathsf{Adv}_{\mathsf{PKE},D}^{\mathsf{ind\text{-}cpa}}(\cdot)$.

∎

F Proof of Theorem 5.1

Let $A = (A_1, A_2)$ be a PT unpredictable adversary attacking F. Let $B = (B_1, B_2)$ be an adversary attacking H as follows. Since B_1 is exactly A_1, and A is unpredictable, B is also unpredictable.

$\underline{B_1(1^\lambda)}$	$\underline{\mathrm{LRSIM}(m_0, m_1, i)}$
$\mathbf{pw} \leftarrow\$\ A_1(1^\lambda)\ ;\ $ Return $\mathbf{pw}$	$x \leftarrow (1^\lambda, \mathbf{sa}[i], \mathbf{hpw}[i])\ ;\ c \leftarrow\$\ \mathsf{WE.Enc}(1^\lambda, x, m_b)$
$\underline{B_2(1^\lambda, \mathbf{sa}, \mathbf{hpw})}$	Return c
$b \leftarrow\$\ \{0,1\}\ ;\ b' \leftarrow\$\ A_2^{\mathrm{LRSIM}}(1^\lambda, \mathbf{sa}, \mathbf{hpw})$	
If $(b = b')$ then return 1 else return 0	

$\underline{B_1(1^\lambda)}$ $\mathbf{pw} \leftarrow\!\!\$\ A_1(1^\lambda)$; Return $\mathbf{pw}$ $\underline{B_2(1^\lambda, \mathbf{sa}, \mathbf{pk})}$ $b \leftarrow\!\!\$\ \{0,1\}$; $b' \leftarrow\!\!\$\ A_2^{\mathrm{LRSIM}}(1^\lambda, \mathbf{sa}, \mathbf{pk})$ Return $(b = b')$ $\underline{\mathrm{LRSIM}(m_0, m_1, i)}$ Return $\mathsf{PKE.Enc}(\mathbf{pk}[i], m_b)$	$\underline{D^{\mathrm{LR}}(1^\lambda, pk)}$ $\mathbf{pw} \leftarrow\!\!\$\ A_1(1^\lambda)$; $s \leftarrow\!\!\$\ \{1, \ldots, p(\lambda)\}$ For $i = 1$ to $\lvert\mathbf{pw}\rvert$ do $\mathbf{sa}[i] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.kl}(\lambda)}$; $(\mathbf{pk}[i], \mathbf{sk}[i]) \leftarrow \mathsf{PKE.Kg}(1^\lambda)$ $\mathbf{pk}[s] \leftarrow pk$; $b' \leftarrow\!\!\$\ A_2^{\mathrm{LRSIM}}(1^\lambda, \mathbf{sa}, \mathbf{pk})$; Return b' $\underline{\mathrm{LRSIM}(m_0, m_1, i)}$ If $i < s$ then return $\mathsf{PKE.Enc}(\mathbf{pk}[i], m_0)$ Else if $i > s$ then return $\mathsf{PKE.Enc}(\mathbf{pk}[i], m_1)$ Else return $\mathrm{LR}(m_0, m_1)$

Figure 13: Adversaries $B = (B_1, B_2)$ and D in the proof of Theorem E.1.

$\underline{D(1^\lambda)}$
$\mathbf{pw} \leftarrow\!\!\$\ A_1(1^\lambda)$; $r \leftarrow\!\!\$\ \{0,1\}^{\rho(\lambda)}$
For $i = 1$ to $\lvert\mathbf{pw}\rvert$ do
  $\mathbf{sa}[i] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.kl}(\lambda)}$; $\mathbf{hpw}[i] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.ol}(\lambda)}$
$j \leftarrow 1$; $s \leftarrow\!\!\$\ \{1, \ldots, q(\lambda)\}$; $A_2^{\mathrm{LRSIM}}(1^\lambda, \mathbf{sa}, \mathbf{hpw}; r)$
$\mathrm{St} \leftarrow (1^\lambda, \mathbf{c}, s, r, \mathbf{sa}, \mathbf{hpw})$
$(x, m_0, m_1) \leftarrow p$; Return $(x, m_0, m_1, \mathrm{St})$
 	
$\underline{D(\mathrm{St}, c)}$
$(1^\lambda, \mathbf{c}, s, r, \mathbf{sa}, \mathbf{hpw}) \leftarrow \mathrm{St}$
$\mathbf{c}[s] \leftarrow c$; $b' \leftarrow A_2^{\mathrm{LRSIM}}(1^\lambda, \mathbf{sa}, \mathbf{hpw}; r)$; Return b'
	$\underline{\mathrm{LRSIM}(m_0, m_1, i)}$
	$x \leftarrow (1^\lambda, \mathbf{sa}[i], \mathbf{hpw}[i])$
	If $j = s$ then
	  $p \leftarrow (x, m_0, m_1)$; $j \leftarrow j + 1$; Return $\mathbf{c}[s]$
	If $j < s$ then $m \leftarrow m_0$ else $m \leftarrow m_1$
	$c \leftarrow\!\!\$\ \mathsf{WE.Enc}(1^\lambda, x, m)$
	If $j < s$ then
	  If $\mathrm{St} = \bot$ then $\mathbf{c}[j] \leftarrow c$ else $c \leftarrow \mathbf{c}[j]$
	$j \leftarrow j + 1$; Return c

Figure 14: **The code of adversary D in Theorem 5.1.**

Next, we'll describe an adversary D attacking WE. Let ρ and q be polynomials that bound the number of coins and the number of oracle queries used by A_2. Adversary $D(1^\lambda)$ runs $A_1(1^\lambda)$ to generate $\mathbf{pw}$. Instead of hashing passwords, adversary D will generate a vector $\mathbf{hpw}$ of uniformly random strings. The assumption that $2^{\mathsf{H.il}(\cdot)-\mathsf{H.ol}(\cdot)}$ is negligible means it's likely that $(1^\lambda, \mathbf{sa}[i], \mathbf{hpw}[i]) \notin L_{\mathsf{H}}$ for every $i \le \lvert\mathbf{hpw}\rvert$. Recall that A may make several oracle queries but D is allowed only a single query $(x, m_0, m_1, \mathrm{St})$. To resolve this, we use the following hybrid argument. Let D pick a random index $s \leftarrow\!\!\$\ \{1, \ldots, q(\lambda)\}$. For the j-th query (m_0, m_1, i) of A, if $j = s$ then D produces its own query $(x, m_0, m_1, \mathrm{St})$, with $x = (1^\lambda, \mathbf{sa}[i], \mathbf{hpw}[i])$, and then later returns its given ciphertext to A. Otherwise, D returns $\mathsf{WE.Enc}(1^\lambda, (1^\lambda, \mathbf{sa}[i], \mathbf{hpw}[i]), m)$, with $m = m_0$ if $j < s$, and $m = m_1$ if $j > s$. Finally, it outputs A's guess b'. The code of D is specified in Fig. 14.

Consider games H_1 and H_2 below, in which game H_2 includes the boxed statement but game H_1 doesn't.

$\underline{\textsc{Game}\ H_1^A(\lambda), \boxed{H_2^A(\lambda)}}$
$\mathbf{pw} \leftarrow\!\!\$\ A_1(1^\lambda)$; $b \leftarrow\!\!\$\ \{0,1\}$
For $i = 1$ to $\lvert\mathbf{pw}\rvert$ do
  $\mathbf{sa}[i] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.kl}(\lambda)}$
  $\mathbf{hpw}[i] \leftarrow \mathsf{H}(1^\lambda, \mathbf{sa}[i], \mathbf{pw}[i])$; $\boxed{\mathbf{hpw}[i] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.ol}(\lambda)}}$
$b' \leftarrow\!\!\$\ A_2^{\mathrm{LR}}(1^\lambda, \mathbf{sa}, \mathbf{hpw})$
Return $(b = b')$
	$\underline{\mathrm{LR}(m_0, m_1, i)}$
	$x \leftarrow (1^\lambda, \mathbf{sa}[i], \mathbf{hpw}[i])$
	Return $\mathsf{WE.Enc}(1^\lambda, x, m_b)$

On the one hand,

$$\begin{aligned}\Pr[\,\mathrm{RIP}^B_{\mathsf{H}}(\lambda) \Rightarrow \mathsf{true} \,|\, a=1\,] &= \Pr[H^A_1(\lambda)] = \Pr[\mathrm{APBE}^A_{\mathsf{F}}(\lambda)], \text{ and} \\ \Pr[\,\mathrm{RIP}^B_{\mathsf{H}}(\lambda) \Rightarrow \mathsf{false} \,|\, a=0\,] &= \Pr[H^A_2(\lambda)]\end{aligned}$$

for every $\lambda \in \mathbb{N}$, where a is the challenge bit of game $\mathrm{RIP}^B_{\mathsf{H}}$. On the other hand, we claim that

$$\begin{aligned}& 2\Pr[\mathrm{AS}^D_{\mathsf{WE},L_{\mathsf{H}}}(\lambda)] - 1 \\ \geq\; & \frac{1}{q(\lambda)}(\Pr[\,H^A_2(\lambda) \Rightarrow \mathsf{true} \,|\, d=1\,] - \Pr[\,H^A_2(\lambda) \Rightarrow \mathsf{false} \,|\, d=0\,]) - 2^{\mathsf{H.il}(\lambda)-\mathsf{H.ol}(\lambda)+1},\end{aligned}$$

for every $\lambda \in \mathbb{N}$, where d is the challenge bit b that game H^A_2 samples. Summing up, $\mathsf{Adv}^{\mathsf{apbe}}_{\mathsf{F},A}(\lambda) \leq 2\mathsf{Adv}^{\mathsf{rip}}_{\mathsf{H},B}(\lambda) + q(\lambda)\cdot\mathsf{Adv}^{\mathsf{as}}_{\mathsf{WE},L_{\mathsf{H}},D}(\lambda) + q(\lambda)\cdot 2^{\mathsf{H.il}(\lambda)-\mathsf{H.ol}(\lambda)+1}$, for every $\lambda \in \mathbb{N}$, and thus F is a secure A-PBE scheme. To justify the claim above, consider the following games G_s, P_s, for $s \in \{1,\ldots,q(\lambda)\}$, in which each game P_s contains the corresponding boxed statement, but game G_s does not.

Game $G^A_s(\lambda)$, $\boxed{P^A_s(\lambda)}$	$\mathrm{LR}(m_0, m_1, i)$
$\mathbf{pw} \leftarrow\!\$\; A_1(1^\lambda)$; $b \leftarrow\!\$\; \{0,1\}$; $\mathsf{passed} \leftarrow \mathsf{true}$	$x \leftarrow (1^\lambda, \mathbf{sa}[i], \mathbf{hpw}[i])$; $m \leftarrow m_b$
For $i = 1$ to $\lvert\mathbf{pw}\rvert$ do	If $j = s$ then
$\mathbf{sa}[i] \leftarrow\!\$\; \{0,1\}^{\mathsf{H.kl}(\lambda)}$; $\mathbf{hpw}[i] \leftarrow\!\$\; \{0,1\}^{\mathsf{H.ol}(\lambda)}$	If $x \in L_{\mathsf{H}}$ then $\mathsf{bad} \leftarrow \mathsf{true}$; $\boxed{\mathsf{passed} \leftarrow \mathsf{false}}$
$j \leftarrow 1$; $b' \leftarrow\!\$\; A_2^{\mathrm{LR}}(1^\lambda, \mathbf{sa}, \mathbf{hpw})$	If $j < s$ then $m \leftarrow m_0$ elsif $j > s$ then $m \leftarrow m_1$
Return $(b = b') \wedge \mathsf{passed}$	$j \leftarrow j+1$; Return $\mathsf{WE.Enc}(1^\lambda, x, m)$

For each $s \in \{1,\ldots,q(\lambda)\}$, games G^A_s and P^A_s are identical-until-bad, and from the Fundamental Lemma of game playing [10],

$$\Pr[G^A_s(\lambda)] - \Pr[P^A_s(\lambda)] \leq \Pr[G^A_s(\lambda) \text{ sets } \mathsf{bad}] \leq 2^{\mathsf{H.il}(\lambda)-\mathsf{H.ol}(\lambda)}$$

for every $\lambda \in \mathbb{N}$; the last inequality is due the fact that, for each fixed $\lambda \in \mathbb{N}$ and $sa \in \{0,1\}^{\mathsf{H.kl}(\lambda)}$, the set $\{\, (1^\lambda, sa, \mathsf{H}(1^\lambda, sa, pw)) \;:\; pw \in \{0,1\}^{\mathsf{H.il}(\lambda)} \,\}$ contains at most $2^{\mathsf{H.il}(\lambda)}$ elements. Let b_s be the challenge bit b that game G^A_s samples. Then $\Pr[\,G^A_s(\lambda) \Rightarrow \mathsf{true} \,|\, b_s = 1\,] = \Pr[\,G^A_{s-1}(\lambda) \Rightarrow \mathsf{false} \,|\, b_{s-1} = 0\,]$ for every $s \in \{2,3,\ldots,q(\lambda)\}$ and every $\lambda \in \mathbb{N}$, and thus

$$\begin{aligned}\sum_{s=1}^{q(\lambda)}\big(2\Pr[G^A_s(\lambda)] - 1\big) &= \sum_{s=1}^{q(\lambda)}\big(\Pr[\,G^A_s(\lambda) \Rightarrow \mathsf{true} \,|\, b_s=1\,] - \Pr[\,G^A_s(\lambda) \Rightarrow \mathsf{false} \,|\, b_s=0\,]\big) \\ &= \Pr[\,G^A_1(\lambda) \Rightarrow \mathsf{true} \,|\, b_1=1\,] - \Pr[\,G^A_{q(\lambda)}(\lambda) \Rightarrow \mathsf{false} \,|\, b_{q(\lambda)}=0\,] \\ &= \Pr[\,H^A_2(\lambda) \Rightarrow \mathsf{true} \,|\, d=1\,] - \Pr[\,H^A_2(\lambda) \Rightarrow \mathsf{false} \,|\, d=0\,] \qquad (2)\end{aligned}$$

for every $\lambda \in \mathbb{N}$. Moreover,

$$\begin{aligned}-1 + 2\cdot\Pr[\mathrm{AS}^D_{\mathsf{WE},L_{\mathsf{H}}}(\lambda)] &= -1 + \frac{2}{q(\lambda)}\sum_{s=1}^{q(\lambda)}\Pr[P^A_s(\lambda)] \geq -1 + \frac{2}{q(\lambda)}\sum_{s=1}^{q(\lambda)}\big(\Pr[G^A_s(\lambda)] - 2^{\mathsf{H.il}(\lambda)-\mathsf{H.ol}(\lambda)}\big) \\ &= -2^{\mathsf{H.il}(\lambda)-\mathsf{H.ol}(\lambda)+1} + \frac{1}{q(\lambda)}\sum_{s=1}^{q(\lambda)}\big(2\Pr[G^A_s(\lambda)] - 1\big) \qquad (3)\end{aligned}$$

for every $\lambda \in \mathbb{N}$. From Equations (2) and (3), the claim follows.

G Proof of Theorem 5.3

Let $A = (A_1, A_2)$ be a PT unpredictable adversary attacking F. Let ρ and q be polynomials that bound the number of coins and the number of oracle queries used by A_2. We'll construct an adversary D attacking WE. Adversary $D(1^\lambda)$ runs $A_1(1^\lambda)$ to generate $\mathbf{pw}$, and hashes these passwords to produce $\mathbf{hpw}$. Recall that A may make several oracle queries but D is allowed only a single query $(x, m_0, m_1, \mathrm{St})$. To resolve this, we use the following hybrid argument. Let D pick a random index $s \leftarrow\$ \{1, \ldots, q(\lambda)\}$. For the j-th query (m_0, m_1, i) of A, if $j = s$ then D produces its own query $(x, m_0, m_1, \mathrm{St})$, with $x = (1^\lambda, \mathbf{sa}[i], \mathbf{hpw}[i])$, and then later returns its given ciphertext to A. Otherwise, D returns $\mathsf{WE.Enc}(1^\lambda, (1^\lambda, \mathbf{sa}[i], \mathbf{hpw}[i]), m)$, with $m = m_0$ if $j < s$, and $m = m_1$ if $j > s$. Finally, it outputs A's guess b'. The code of D is shown below.

$\underline{D(1^\lambda)}$
$\mathbf{pw} \leftarrow\$ A_1(1^\lambda)$; $r \leftarrow\$ \{0,1\}^{\rho(\lambda)}$
For $i = 1$ to $|\mathbf{pw}|$ do
  $\mathbf{sa}[i] \leftarrow\$ \{0,1\}^{\mathsf{H.kl}(\lambda)}$; $\mathbf{hpw}[i] \leftarrow \mathsf{H}(1^\lambda, \mathbf{sa}[i], \mathbf{pw}[i])$
$j \leftarrow 1$; $s \leftarrow\$ \{1, \ldots, q(\lambda)\}$; $A_2^{\mathrm{LRSIM}}(1^\lambda, \mathbf{sa}, \mathbf{hpw}; r)$
$\mathrm{St} \leftarrow (1^\lambda, \mathbf{c}, s, r, \mathbf{sa}, \mathbf{hpw})$
$(x, m_0, m_1) \leftarrow p$; Return $(x, m_0, m_1, \mathrm{St})$

$\underline{D(\mathrm{St}, c)}$
$(1^\lambda, \mathbf{c}, s, r, \mathbf{sa}, \mathbf{hpw}) \leftarrow \mathrm{St}$
$\mathbf{c}[s] \leftarrow c$; $b' \leftarrow A_2^{\mathrm{LRSIM}}(1^\lambda, \mathbf{sa}, \mathbf{hpw}; r)$; Return b'

$\underline{\mathrm{LRSIM}(m_0, m_1, i)}$
$x \leftarrow (1^\lambda, \mathbf{sa}[i], \mathbf{hpw}[i])$
If $j = s$ then
  $p \leftarrow (x, m_0, m_1)$; $j \leftarrow j + 1$; Return $\mathbf{c}[s]$
If $j < s$ then $m \leftarrow m_0$ else $m \leftarrow m_1$
$c \leftarrow\$ \mathsf{WE.Enc}(1^\lambda, x, m)$
If $j < s$ then
  If $\mathrm{St} = \perp$ then $\mathbf{c}[j] \leftarrow c$ else $c \leftarrow \mathbf{c}[j]$
$j \leftarrow j + 1$; Return c

Let $\mathsf{R_H}$ be the **NP**-relation of L_{H}, that is, $\mathsf{R_H}\big((1^\lambda, sa, hpw), pw\big)$ returns $(\mathsf{H}(1^\lambda, sa, pw) = hpw)$. Since D is PT and WE is XS$[\mathsf{R_H}]$-secure, there exists a PT extractor E such that $\mathsf{Adv}^{\mathsf{xs}}_{\mathsf{WE},\mathsf{R_H},D,E}(\cdot)$ is negligible. Construct $B = (B_1, B_2)$ attacking H as follows. Since B_1 is exactly A_1, and A is unpredictable, B is also unpredictable.

$\underline{B_1(1^\lambda)}$
$\mathbf{pw} \leftarrow\$ A_1(1^\lambda)$; Return $\mathbf{pw}$

$\underline{B_2(1^\lambda, \mathbf{sa}, \mathbf{hpw})}$
$b \leftarrow\$ \{0,1\}$; $j \leftarrow 1$; $r \leftarrow\$ \{0,1\}^{\rho(\lambda)}$
$s \leftarrow\$ \{1, \ldots, q(\lambda)\}$; $A_2^{\mathrm{LRSIM}}(1^\lambda, \mathbf{sa}, \mathbf{hpw}; r)$
$(w, i) \leftarrow p$; Return (w, i)

$\underline{\mathrm{LRSIM}(m_0, m_1, i)}$
$x \leftarrow (1^\lambda, \mathbf{sa}[i], \mathbf{hpw}[i])$; $m \leftarrow m_b$
If $j < s$ then $m \leftarrow m_0$ elsif $j > s$ then $m \leftarrow m_1$
$c \leftarrow\$ \mathsf{WE.Enc}(1^\lambda, x, m)$
If $j = s$ then
  $\mathrm{St} \leftarrow (1^\lambda, \mathbf{c}, s, r, \mathbf{sa}, \mathbf{hpw})$
  $w \leftarrow\$ E(1^\lambda, x, m_0, m_1, \mathrm{St}, c)$; $p \leftarrow (w, i)$
$\mathbf{c}[j] \leftarrow c$; $j \leftarrow j + 1$; Return c

Consider the following games G_s, for $s \in \{1, \ldots, q(\lambda)\}$.

$\underline{\text{GAME } G_s^{A,E}(\lambda)}$
$\mathbf{pw} \leftarrow\$ A_1(1^\lambda)$; $b \leftarrow\$ \{0,1\}$; $\mathsf{passed} \leftarrow \mathsf{false}$
For $i = 1$ to $|\mathbf{pw}|$ do
  $\mathbf{sa}[i] \leftarrow\$ \{0,1\}^{\mathsf{H.kl}(\lambda)}$
  $\mathbf{hpw}[i] \leftarrow \mathsf{H}(1^\lambda, \mathbf{sa}[i], \mathbf{pw}[i])$
$j \leftarrow 1$; $r \leftarrow\$ \{0,1\}^{\rho(\lambda)}$
$b' \leftarrow A_2^{\mathrm{LR}}(1^\lambda, \mathbf{sa}, \mathbf{hpw}; r)$
Return $(b = b')$

$\underline{\mathrm{LR}(m_0, m_1, i)}$
$x \leftarrow (1^\lambda, \mathbf{sa}[i], \mathbf{hpw}[i])$; $m \leftarrow m_b$
If $j < s$ then $m \leftarrow m_0$ elsif $j > s$ then $m \leftarrow m_1$
$c \leftarrow\$ \mathsf{WE.Enc}(1^\lambda, x, m)$
If $j = s$ then
  $\mathrm{St} \leftarrow (1^\lambda, \mathbf{c}, s, r, \mathbf{sa}, \mathbf{hpw})$
  $w \leftarrow\$ E(1^\lambda, x, m_0, m_1, \mathrm{St}, c)$
  If $(\mathsf{H}(1^\lambda, \mathbf{sa}[i], w) = \mathbf{hpw}[i])$ then $\mathsf{passed} \leftarrow \mathsf{true}$
$\mathbf{c}[j] \leftarrow c$; $j \leftarrow j + 1$; Return c

Let P_s^A and H_s^A be identical to G_s^A, with the following difference: game P_s^A returns passed and game H_s^A returns $(b = b') \wedge \neg\mathsf{passed}$. Let b_s be the challenge bit b that game $G_s^{A,E}$ samples. Then

$$\Pr[\, G_s^{A,E}(\cdot) \Rightarrow \mathsf{true} \mid b_s = 1\,] = \Pr[\, G_{s-1}^{A,E}(\cdot) \Rightarrow \mathsf{false} \mid b_{s-1} = 0\,]$$

for every $s \in \{2, 3, \ldots, q\}$, and thus

$$\begin{aligned}
\sum_{s=1}^{q}\bigl(2\Pr[G_s^{A,E}(\cdot)] - 1\bigr) &= \sum_{s=1}^{q}\bigl(\Pr[\, G_s^{A,E}(\cdot) \Rightarrow \mathsf{true} \mid b_s = 1\,] - \Pr[\, G_s^{A,E}(\cdot) \Rightarrow \mathsf{false} \mid b_s = 0\,]\bigr) \\
&= \Pr[\, G_1^{A,E}(\cdot) \Rightarrow \mathsf{true} \mid b_1 = 1\,] - \Pr[\, G_q^{A,E}(\cdot) \Rightarrow \mathsf{false} \mid b_q = 0\,] \\
&= \Pr[\, \mathrm{APBE}_{\mathsf{F}}^{A}(\cdot) \Rightarrow \mathsf{true} \mid d = 1\,] - \Pr[\, \mathrm{APBE}_{\mathsf{F}}^{A}(\cdot) \Rightarrow \mathsf{false} \mid d = 0\,] \\
&= \mathsf{Adv}_{\mathsf{F},A}^{\mathsf{apbe}}(\cdot),
\end{aligned}$$

where d is the challenge bit of game $\mathrm{APBE}_{\mathsf{F}}$. Moreover,

$$\begin{aligned}
-1 + 2\Pr[\mathrm{XS}_{\mathsf{WE},\mathsf{R_H}}^{A,E}(\cdot)] &= -1 + \frac{2}{q}\sum_{s=1}^{q}\Pr[H_s^{A,E}(\cdot)] \\
&\geq -1 + \frac{2}{q}\sum_{s=1}^{q}\bigl(\Pr[G_s^{A,E}(\cdot)] - \Pr[P_s^{A,E}(\cdot)]\bigr) \\
&= -\frac{2}{q}\sum_{s=1}^{q}\Pr[P_s^{A,E}(\cdot)] + \frac{1}{q}\sum_{s=1}^{q}\bigl(2\Pr[G_s^{A,E}(\cdot)] - 1\bigr) \\
&= -2\Pr[\mathrm{ROW}_{\mathsf{H}}^{B}(\cdot)] + \frac{1}{q}\sum_{s=1}^{q}\bigl(2\Pr[G_s^{A,E}(\cdot)] - 1\bigr) \\
&= -2\mathsf{Adv}_{\mathsf{H},B}^{\mathsf{row}}(\cdot) + \frac{1}{q}\mathsf{Adv}_{\mathsf{F},A}^{\mathsf{apbe}}(\cdot) \; .
\end{aligned}$$

Hence, $\mathsf{Adv}_{\mathsf{F},A}^{\mathsf{apbe}}(\cdot) \leq 2q \cdot \mathsf{Adv}_{\mathsf{H},B}^{\mathsf{row}}(\cdot) + q \cdot \mathsf{Adv}_{\mathsf{WE},\mathsf{R_H},D,E}^{\mathsf{xs}}(\cdot)$, and thus F is a secure A-PBE scheme.

H Proof of Proposition 5.2

Assume we are given $\mathsf{WE} \in \mathsf{XS}[\mathcal{L}(\mathsf{R})]$. We want to show that WE is $\mathsf{AS}[\mathcal{L}(\mathsf{R})]$-secure. Let A be a PT adversary. Then, there is a PT extractor E such that $\mathsf{Adv}_{\mathsf{WE},\mathsf{R},A,E}^{\mathsf{xs}}(\cdot)$ is negligible. Consider the following games H_1 and H_2; the latter includes the boxed statement but the former does not.

$$\begin{array}{l}
\underline{\textsc{Game}\ H_1^{A,E}(\lambda),\ \boxed{H_2^{A,E}(\lambda)}} \\
(x, m_0, m_1, \mathrm{St}) \leftarrow\!\$\; A(1^\lambda)\; ;\; b \leftarrow\!\$\; \{0,1\} \\
c \leftarrow\!\$\; \mathsf{WE.Enc}(1^\lambda, x, m_b)\; ;\; b' \leftarrow\!\$\; A(\mathrm{St}, c) \\
w \leftarrow\!\$\; E(1^\lambda, x, m_0, m_1, \mathrm{St}, c) \\
\boxed{\text{If } (x \in \mathcal{L}(\mathsf{R})) \wedge \neg\mathsf{R}(x,w) \text{ then return } \mathsf{false}} \\
\text{Return } (b = b') \wedge \neg\mathsf{R}(x,w)
\end{array}$$

On the one hand, $\Pr[H_1^{A,E}(\cdot)] = \Pr[\mathrm{XS}_{\mathsf{WE},\mathsf{R}}^{A,E}(\cdot)]$ and $\Pr[H_2^{A,E}(\cdot)] = \Pr[\mathrm{AS}_{\mathsf{WE},\mathcal{L}(\mathsf{R})}^{A}(\cdot)]$. On the other hand, $\Pr[H_1^{A,E}(\cdot)] \geq \Pr[H_2^{A,E}(\cdot)]$. Hence $\mathsf{Adv}_{\mathsf{WE},\mathcal{L}(\mathsf{R}),A}^{\mathsf{as}}(\cdot) \leq \mathsf{Adv}_{\mathsf{WE},\mathsf{R},A,E}^{\mathsf{xs}}(\cdot)$, and thus $\mathsf{WE} \in \mathsf{AS}[\mathcal{L}(\mathsf{R})]$.

I Constructing XS-secure WE

We begin by giving an alternative definition for the recent notion of extractability obfuscator, equivalently differing-input obfuscator [4, 13, 1].

Extractability obfuscators. Let P be an obfuscator, defining a PT obfuscation algorithm $\mathsf{P.Ob}$ and a PT evaluation algorithm $\mathsf{P.Ev}$. We say that P is xO-secure if for every PT adversary A, there is a PT algorithm (extractor) E such that $\mathsf{Adv}^{\mathsf{xo}}_{\mathsf{P},A,E}(\lambda) = 2\Pr[\mathrm{XO}^{A,E}_{\mathsf{P}}(\lambda)] - 1$ is negligible, where game XO is defined at as follows:

$$
\begin{array}{l}
\underline{\textsc{Game}\ \mathrm{XO}^{A,E}_{\mathsf{P}}(\lambda)} \\
(C_0, C_1, \mathrm{St}) \leftarrow\!\!\$\ A(1^\lambda)\ ;\ b \leftarrow\!\!\$\ \{0,1\}\ ;\ c \leftarrow\!\!\$\ \mathsf{P.Ob}(1^\lambda, C_b) \\
b' \leftarrow\!\!\$\ A(\mathrm{St}, c)\ ;\ w \leftarrow\!\!\$\ E(1^\lambda, C_0, C_1, \mathrm{St}, c) \\
\text{Return } (b = b') \wedge (C_0(w) = C_1(w))
\end{array}
$$

In the game above, circuits C_0, C_1 must have the same size.

Achieving XS security. Recall that in Section 3, we have the construction $\mathsf{WE}_{\mathsf{R}}[\mathsf{P}]$ of witness encryption for language $\mathcal{L}(\mathsf{R}) \in \mathbf{NP}$ from obfuscator P. The following says that if P is assumed to be xO-secure then $\mathsf{WE}_{\mathsf{R}}[\mathsf{P}]$ is $\mathrm{XS}[\mathcal{L}(\mathsf{R})]$-secure.

Theorem I.1 Let R be an $\mathbf{NP}$ relation, and let P be an obfuscator. Construct $\mathsf{WE}_{\mathsf{R}}[\mathsf{P}]$ as in Section 3. If P is xO-secure then $\mathsf{WE}_{\mathsf{R}}[\mathsf{P}] \in \mathsf{XS}[\mathcal{L}(\mathsf{R})]$.

Proof: For each $x, m \in \{0,1\}^*$, let $R_{x,m}$ be a circuit that on input $w \in \{0,1\}^{\mathsf{R.wl}(|x|)}$, returns m if $\mathsf{R}(x, w)$ and returns $0^{|m|}$ otherwise. Let A be a PT adversary attacking $\mathsf{WE}_{\mathsf{R}}[\mathsf{P}]$. Wlog, assume that A produces distinct m_0 and m_1. Consider the following adversary B attacking P.

$$
\begin{array}{l|l}
\underline{B(1^\lambda)} & \underline{B(\mathrm{St}, c)} \\
(x, m_0, m_1, \mathrm{St}) \leftarrow\!\!\$\ A(1^\lambda)\ ;\ \text{Return } (R_{x,m_0}, R_{x,m_1}, \mathrm{St}) & b' \leftarrow\!\!\$\ A(\mathrm{St}, c)\ ;\ \text{Return } b'
\end{array}
$$

Since B is PT and P is xO-secure, there is a PT extractor E such that $\mathsf{Adv}^{\mathsf{xo}}_{\mathsf{P},B,E}(\cdot)$ is negligible. Consider the following extractor $\overline{E}$ for A:

$$
\begin{array}{l}
\underline{\overline{E}(1^\lambda, x, m_0, m_1, \mathrm{St}, c)} \\
w \leftarrow\!\!\$\ E(1^\lambda, R_{x,m_0}, R_{x,m_1}, \mathrm{St}, c)\ ;\ \text{Return } w
\end{array}
$$

This extractor $\overline{E}$ is PT. Note that for any $w \in \{0,1\}^{\mathsf{R.wl}(x)}$, we have $R_{x,m_0}(w) \neq R_{x,m_1}(w)$ if and only if $\mathsf{R}(x, w)$. Then $\Pr[\mathrm{XS}^{A,\overline{E}}_{\mathsf{WE}_{\mathsf{R}}[\mathsf{P}],\mathsf{R}}(\cdot)] = \Pr[\mathrm{XO}^{B,E}_{\mathsf{P}}(\cdot)]$, and thus $\mathsf{Adv}^{\mathsf{xs}}_{\mathsf{WE}_{\mathsf{R}}[\mathsf{P}],\mathsf{R},A,\overline{E}}(\cdot) = \mathsf{Adv}^{\mathsf{xo}}_{\mathsf{P},B,E}(\cdot)$. Hence $\mathsf{WE}_{\mathsf{R}}[\mathsf{P}]$ is $\mathrm{XS}[\mathcal{L}(\mathsf{R})]$-secure. ▮