Adaptive Witness Encryption and Asymmetric Password-based Cryptography

Mihir Bellare[1] Viet Tung Hoang[2]

October 17, 2013

Abstract

This paper defines adaptive soundness (AS) security for witness encryption and applies it to provide the first non-invasive schemes for asymmetric password-based encryption (A-PBE). A-PBE offers significant gains over classical, symmetric password-based encryption (S-PBE) in the face of attacks that compromise servers to recover hashed passwords. We also show by counter-example that the original soundness security (SS) requirement of GGSW does not suffice for the security of their own applications, and show that AS fills the gap.

[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-0904380, CCF-0915675, CNS-1116800 and CNS-1228890.

[2] Department of Computer Science & Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. Email: vth005@eng.ucsd.edu. URL: http://csiflabs.cs.ucdavis.edu/~tvhoang/. Supported in part by NSF grants CNS-0904380, CCF-0915675, CNS-1116800 and CNS-1228890.

Contents

1 Introduction

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

The problem. Today secure Internet communication remains ubiquitously based on client passwords. Standards such as the widely implemented PKCS#5 (equivalently, RFC 2898) [18] specify password-based encryption (PBE). From the client password pw, one derives a hashed password $hpw = \mathsf{KD}(sa, pw)$, where sa is a random, user-specific public salt, and KD, the deterministic key-derivation function, is usually an iterated hash, $\mathsf{KD}(sa, pw) = H^t(sa|pw)$ for some iteration count t and 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. We have seen a series of high-profile attacks of this type, including Adobe (October 2013), LinkedIn (June 2012), RSA (March 2011), Sony (2011) and TJ Maxx (2007). 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 with symmetric PBE (S-PBE), possession of the hashed password is already enough to decrypt any prior communications, meaning even well-chosen passwords, not subject to dictionary attack, do not provide security in the face of server compromise.

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

A-PBE is trivial to achieve if we have the luxury of designing our own KD. Namely, let KD, given sa, pw, deterministically derive from pw a string r of coin tosses for a key-generation algorithm $\mathsf{PKE.Kg}$ of some standard PKE scheme. It then runs $\mathsf{PKE.Kg}$ on r to get (pk, sk) and outputs $hpw = pk$. Encryption is under the encryption algorithm $\mathsf{PKE.Enc}$ of the PKE scheme keyed with $hpw = pk$. Since KD is deterministic, decryption under (sa, pw) can re-execute KD to get (sk, pk) and then use sk to decrypt under PKE.[1]

From a practical perspective, however, the above is a non-solution. The reason is that it is *invasive*, prescribing a particular and very special way to design KD. Right now, in practice, the key-derivation functions in use do nothing like the KD sketched above. Instead, they are iterated hash functions following standards like PKCS#5 [18]. Millions of passwords are today in use with this particular KD, and we do not have the luxury of changing the password hashes or the KD. In the face of this legacy constraint, the practical problem is to implement APBE in a *non-invasive* way, meaning without changing the key-derivation function KD. In particular, for a A-PBE solution to be useful, it should be able to use as public key a hashed password obtained with KD being an iterated hash function.

Achieving A-PBE non-invasively is much more challenging, and indeed looks almost impossible. In all known PKE scheme, 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 the new witness encryption (WE) primitive introduced by Garg, Gentry, Sahai and Waters (GGSW) [12, 13]. 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 [12, 13] that we call adaptive soundness security. We define and achieve WE with adaptive soundness and apply it to achieve non-invasive A-PBE and other goals as we now discuss.

Witness encryption. In a WE scheme [12, 13] 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

[1] One might ask how KD can deterministically derive a random-looking r from sa, pw. The simplest way is to apply to $sa\|pw$ a cryptographic hash function modeled as a random oracle [5].

x in L. If $x \notin L$ then the message remains private given the ciphertext. The soundness security (SS) requirement of GGSW [12, 13] formalizes the latter by asking that for any PT adversary A, any $x \notin L$ and any equal-length messages m_0, m_1 in the message space, 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 $\lambda \in \mathbb{N}$. GGSW [12, 13] give a construction of SS-secure WE for the **NP**-complete Exact-Cover language based on multi-linear maps [10]. Another construction of SS-secure WE from indistinguishability obfuscation (iO) is given in GGHRSW [11].

Adaptive soundness. We introduce adaptive soundness (AS) security of WE. In our formalization, the adversary A, on input 1^λ, returns 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.

Towards providing candidate AS-secure WE schemes, we return to the (two) known constructions of SS-secure ones. We show that the iO-based WE scheme of GGHRSW [11] is AS-secure, so that we can achieve AS security with no extra assumptions compared with SS security in this case. To show SS security of their Exact-Cover WE scheme, GGSW [12, 13] assume hardness of a new problem they call Decision Multi-linear No-exact-cover. We can obtain AS security of the same scheme assuming hardness of an adaptive version of this problem.

Defining and achieving A-PBE. We provide a definition of the A-PBE goal by extending the S-PBE framework of [4]. Our model involves multiple passwords. They are assumed to individually have high min-entropy, since otherwise security is moot, but they may be arbitrarily related to each other. This reflects the reality that we, as users, pick related passwords, for example varying a base password by appending the name of the website. Our A-PBE scheme lets L be the **NP** language of pairs $(sa, \mathsf{KD}(sa, pw))$ 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 witness encryption of m under $x = hpw$. Decryption will use the witness pw.

The key feature of our solution that distinguishes it from the trivial A-PBE solution outlined above is that ours is non-invasive. It does not prescribe or require any particular design for KD. Rather, it takes KD as given, and shows how to encrypt with public key the hashed password obtained from KD. In this way, KD can in particular be the iterated hash design of the PKCS#5 standard [18] 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 KD, but that is very different from prescribing its design. Our assumption, which we formalize as KDF-pseudorandomness in Section 5, asks that outputs of KD on unpredictable passwords are pseudorandom. 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 [4] to hold for PKCS#5 in the ROM, so it is a natural and reasonable assumption. Note that we do need a standard-model assumption on KD, for if we model it as a RO then the language L we define is not in **NP**.

Due to the inefficiency of existing WE schemes, our A-PBE scheme is not efficient. Our result should be viewed as an indication that non-intrusive A-PBE is achievable in principle. We believe this is significant because of the practical value of the goal and because it is extremely unclear that the goal was achievable, even in principle, prior to WE and our work.

SS revisited. GGSW [12, 13] 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 [12, 13]. The answer is that they need it too. We point out that the theorems of GGSW [12, 13] 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. 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 [12, 13] 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. We do not see the gap in GGSW [12, 13] as serious, since it is easily fixed by moving to AS, but we think the issue is subtle and interesting and accordingly provide the counter-examples in Section 4.

Extractability. Goldwasser, Kalai, Popa, Vaikuntanathan and Zeldovich (GKPVZ) [15, 14] define extractable witness encryption, which says 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 $x \in L$. We build on this to provide a definition of *adaptive* extraction security for WE that we call XS. We show XS implies AS, giving a hierarchy XS $\Rightarrow$ AS $\Rightarrow$ SS. We show that XS allows somewhat stronger results for non-invasive A-PBE than we obtained under AS, namely that the assumptions needed on KD decrease, down to just requiring KD to be one-way. Our XS definition does not allow the adversary and extractor any common auxiliary inputs, which avoids the iO-based attacks of BCPR [7]. Towards achieving XS, we present a construction of an XS-secure WE scheme based on the extractability obfuscation (xO) primitive of BCP [9].

Relaxing perfect correctness in WE. The correctness requirement of WE is that if $x \in L$ then possession of a witness w enables decryption. The simplest formalization is perfect correctness, requiring this to be true with probability one across all choices of inputs involved. GGSW [12, 13] however give a weaker requirement which allows decryption to fail with negligible probability. The motivation appears to be that in the graded encoding rendering of multi-linear maps [10], as opposed to the "dream" version, there are errors that lead to decryption errors in the WE scheme that GGSW build.

We point out that relaxing the correctness requirement is more subtle than it may seem in that the details of how it is done significantly impact the applications of WE. We show that under the most natural interpretation of the GGSW [12, 13] requirement, their PRG-based PKE scheme will fail to provide any correctness at all, meaning decryption will never reverse encryption. Other interpretations of their requirement are possible, but it is currently ambiguously written and it is not clear how to interpret it. We provide our own definition of a relaxation of perfect completeness that suffices for applications. See Appendix A.

Password-based signatures. We have shown how to accomplish (non-invasive) A-PBE. A natural question is whether it is possible to do password-based signatures, meaning we have to sign using pw as the secret key and verify with $hpw = \mathsf{KD}(sa, pw)$ as the public key. Again, this is trivial if we allow an invasive solution, meaning we get to define KD, and the problem of practical interest is non-invasive password-based signatures, meaning we view KD as given rather than as something we construct. We can show how to achieve non-invasive password-based signatures by using key-versatile signatures [3]. The assumption on KD would be an appropriate form of one-wayness.

Summary. The contributions of this work are the notion of adaptive soundness (AS) for witness encryption (WE) and its application to achieve non-invasive, asymmetric password-based encryption (A-PBE). A-PBE offers protection in the face of the swathe of attacks that compromise servers and recover hashed passwords. We have shown that AS-secure WE is strictly stronger than its precursor, SS-secure WE, and that it allows proofs of prior WE-based constructions not possible under SS.

MAIN $\mathrm{INDCPA}^{A}_{\mathsf{PKE}}(\lambda)$	MAIN $\mathrm{PRG}^{A}_{G}(\lambda)$	MAIN $\mathrm{IO}^{A}_{\mathsf{F}}(\lambda)$
$(pk, sk) \leftarrow_{\$} \mathsf{PKE.Kg}(1^\lambda)$; $b \leftarrow_{\$} \{0,1\}$	$s \leftarrow_{\$} \{0,1\}^\lambda$; $x_1 \leftarrow G(s)$	$(C_0, C_1, \mathrm{St}) \leftarrow_{\$} A(1^\lambda)$; $b \leftarrow_{\$} \{0,1\}$
$(m_0, m_1, \mathrm{St}) \leftarrow_{\$} A(1^\lambda, pk)$	$x_0 \leftarrow_{\$} \{0,1\}^{\ell(\lambda)}$; $b \leftarrow_{\$} \{0,1\}$	$c \leftarrow_{\$} \mathsf{F.Ob}(1^\lambda, C_b)$; $b' \leftarrow_{\$} A(\mathrm{St}, c)$
$c \leftarrow_{\$} \mathsf{PKE.Enc}(pk, m_b)$	$b' \leftarrow_{\$} A(1^\lambda, x_b)$; Return $(b = b')$	Return $(b = b') \wedge (C_0 \equiv C_1)$
$b' \leftarrow_{\$} A(1^\lambda, \mathrm{St}, c)$; Return $(b = b')$		

Figure 1: **Left:** Game INDCPA defining INDCPA security of a PKE scheme PKE. The messages $m_0, m_1 \in \mathsf{PKE.Msg}$ must have the same length. **Middle:** Game PRG defining security of a pseudorandom generator G. Here $\ell : \mathbb{N} \to \mathbb{N}$ is the expansion factor of G. **Right:** Game IO defining security of an indistinguishability obfuscator F.

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 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 [6] augmented with explicit MAIN procedures as in [19]. 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 output y, the game output being what is returned by MAIN. 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.

PUBLIC-KEY ENCRYPTION. A public-key encryption (PKE) scheme PKE defines PT algorithms $\mathsf{PKE.Kg}$, $\mathsf{PKE.Enc}$, $\mathsf{PKE.Dec}$, the last deterministic, and an associated message space $\mathsf{PKE.Msg} \subseteq \{0,1\}^*$. Algorithm $\mathsf{PKE.Kg}$ takes as input a unary representation 1^λ of a security parameter λ, and outputs a public key pk and a secret key sk. Algorithm $\mathsf{PKE.Enc}$ takes as input pk and a message $m \in \mathsf{PKE.Msg}$, and outputs a ciphertext c. Algorithm $\mathsf{PKE.Dec}(sk, c)$ outputs $m \in \mathsf{PKE.Msg} \cup \{\bot\}$. Scheme PKE is INDCPA-secure [16, 2] if $\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathsf{PKE},A}(\lambda) = 2[\mathrm{INDCPA}^{A}_{\mathsf{PKE}}(\lambda)] - 1$ is negligible for every PT adversary A, where game INDCPA is defined in the left panel of Fig. 1.

PSEUDORANDOM GENERATORS. A pseudorandom generator (PRG) [8, 21] 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 the middle panel of Fig. 1.

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 F defines PT algorithms $\mathsf{F.Ob}, \mathsf{F.Ev}$. Algorithm $\mathsf{F.Ob}$ takes as input the unary representation 1^λ of a security parameter λ and a circuit C, and outputs a string c. Algorithm $\mathsf{F.Ev}$ takes as input strings c, x and returns $y \in \{0,1\}^* \cup \{\bot\}$. We require that for any circuit C, any input x, and any $\lambda \in \mathbb{N}$, it holds that $\mathsf{F.Ev}(\mathsf{F.Ob}(1^\lambda, C), x) = C(x)$. We say that F is iO-secure if $\mathsf{Adv}^{\mathsf{io}}_{\mathsf{F},A}(\lambda) = 2\Pr[\mathrm{IO}^{A}_{\mathsf{F}}(\lambda)] - 1$ is negligible for every PT adversary A, where game IO is defined at the right panel of Fig. 1. This definition is slightly different from the notion in [1, 11]—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.

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

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

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

3 Adaptive Witness Encryption

We begin by recalling the notion of witness encryption of GGSW [12, 13] and its soundness security requirement. We then present our adaptive definition and show that it is a strictly stronger requirement.

NP RELATIONS. For $\mathsf{R}\colon \{0,1\}^* \times \{0,1\}^* \to \{\mathsf{true}, \mathsf{false}\}$, 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 PT and there is a polynomial $\mathsf{R.wl}\colon \mathbb{N} \to \mathbb{N}$, 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 WE.Enc, WE.Dec and an associated message space $\mathsf{WE.Msg} \subseteq \{0,1\}^*$. Algorithm 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 $m \in \mathsf{WE.Msg}$, and outputs a ciphertext c. Algorithm WE.Dec takes as input a ciphertext c and a string w, and outputs $m \in \mathsf{WE.Msg} \cup \{\bot\}$. Correctness requires that $\mathsf{WE.Dec}(\mathsf{WE.Enc}(1^\lambda, x, m), w) = m$ for all $\lambda \in \mathbb{N}$, all $x \in L$, all $w \in \mathsf{R}(x)$ and all $m \in \mathsf{WE.Msg}$.[2]

SOUNDNESS SECURITY. The soundness security (SS) condition of GGSW [12, 13] says that for any PT adversary A, any $x \in \{0,1\}^* \setminus L$ and any equal-length $m_0, m_1 \in \mathsf{WE.Msg}$, there is a negligible function ν such that for all $\lambda \in \mathbb{N}$ we have

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

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 $m_0, m_1 \in \mathsf{WE.Msg}$, the 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 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 $\mathsf{WE.Enc}(1^\lambda, x, m_b)$. 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.

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

[2] This is perfect correctness. GGSW [12] give a weaker correctness condition in which decryption can sometimes fail. We revisit correctness in Appendix A and show that if one relaxes perfect correctness, one has to be careful, for seemingly minor details of how it is done are crucial for the correctness of applications. For simplicity, we restrict attention to perfect correctness for now.

```
$\mathsf{WE}_f.\mathsf{Enc}(1^\lambda, x, m)$
If $|x| \geq f(\lambda)$ then return $(0, m)$
Else return $(1, \mathsf{WE}.\mathsf{Enc}(1^\lambda, x, m))$
```

```
$\mathsf{WE}_f.\mathsf{Dec}(c, w)$
$(b, t) \leftarrow c$
If $b = 0$ then return $t$ else return $\mathsf{WE}.\mathsf{Dec}(t, w)$
```

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 $f : \mathbb{N} \to \mathbb{N}$. We let $\mathsf{WE}_f.\mathsf{Msg} = \mathsf{WE}.\mathsf{Msg}$.

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 $m_0, m_1 \in \mathsf{WE}.\mathsf{Msg}$ have equal length. Let PT adversary B, on input ciphertext c, return $b' \leftarrow A((1, c))$. Let $S(x) = \{ \lambda \in \mathbb{N} \, : \, f(\lambda) \leq |x| \}$. 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 $m_0, m_1 \in \mathsf{WE}.\mathsf{Msg}$ have equal length. We want to show that the function $\mathsf{Adv}^{\mathsf{ss}}_{\mathsf{WE},R,x,m_0,m_1,A}(\cdot)$ 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 $d \geq 1$ 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 $m_0, m_1 \in \mathsf{WE.Msg}$ be arbitrary, distinct, equal-length messages. Consider the following adversary A:

```
$\underline{A(1^\lambda)}$
$k \leftarrow f(\lambda)$ ; $x \leftarrow 0^k$
For all $s \in \{0,1\}^k$ do
    If $(M(s) \neq 1)$ then $x \leftarrow s$
Return $(x, m_0, m_1, \varepsilon)$
```

```
$\underline{A(t, c)}$
$(b, m) \leftarrow c$
If $((b = 0) \wedge (m = m_1))$ then return 1
Return 0
```

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 $\lambda \in \mathbb{N}$ such that $\{0,1\}^\lambda \setminus L \neq \emptyset$ we will have $\mathsf{Adv}^{\mathsf{as}}_{\mathsf{WE}_f,L,A}(\lambda) = 1$. Since $\{0,1\}^* \setminus L$ is infinite, 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. ▮

Achieving AS-security. Our preferred construction uses indistinguishability obfuscation (iO) [1, 11]. GGHRSW [11] present an iO-based SS[L]-secure WE scheme for any $L \in \mathbf{NP}$. We show that their scheme achieves our stronger AS[L]-security notion under the same assumption. Proceeding to the details, let R be an $\mathbf{NP}$-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 F be an indistinguishability obfuscator, defining a PT obfuscation algorithm $\mathsf{F.Ob}$ and a PT evaluation algorithm $\mathsf{F.Ev}$. We define WE scheme $\mathsf{WE_R[F]}$ as follows: $\mathsf{WE_R[F].Enc}(1^\lambda, x, m)$ returns $c \leftarrow\!\!\$\ \mathsf{F.Ob}(1^\lambda, R_{x,m})$; $\mathsf{WE_R[F].Dec}(c, w)$ returns $m \leftarrow\!\!\$\ \mathsf{F.Ev}(c, w)$; and $\mathsf{WE_R[F].Msg} = \{0,1\}$.

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

Proof: Let A be a PT adversary attacking the AS[L]-security of $\mathsf{WE_R[F]}$. Note that if $x \notin L$ then $R_{x,m} \equiv R_{x,0}$ for any $m \in \{0,1\}$, meaning these two circuits are functionally equivalent. Consider the following PT adversary B attacking F:

```
$\underline{B(1^\lambda)}$
$(x, m_0, m_1, \mathrm{St}) \leftarrow\!\!\$\ A(1^\lambda)$ ; $b \leftarrow\!\!\$\ \{0,1\}$
$t \leftarrow (\mathrm{St}, b)$ ; Return $(R_{x,0}, R_{x,m_b}, t)$
```

```
$\underline{B(t, c)}$
$(\mathrm{St}, b) \leftarrow t$ ; $b' \leftarrow\!\!\$\ A(\mathrm{St}, c)$
If $(b = b')$ then return 1 else return 0
```

Then

$$\Pr[\,\mathrm{AS}^B_{\mathsf{WE_R[F]},L}(\cdot) \Rightarrow \mathsf{true} \,|\, a = 1\,] = \Pr[\mathrm{IO}^A_{\mathsf{F}}(\cdot)] \quad \text{and} \quad \Pr[\,\mathrm{AS}^B_{\mathsf{WE_R[F]},L}(\cdot) \Rightarrow \mathsf{false} \,|\, a = 0\,] = \frac{1}{2},$$

where a is the challenge bit of game $\mathrm{AS}^B_{\mathsf{WE_R[F]}}$. Subtracting, we get $\mathsf{Adv}^{\mathsf{as}}_{\mathsf{WE_R[F]},L,B}(\cdot) = \frac{1}{2}\mathsf{Adv}^{\mathsf{io}}_{\mathsf{F},A}(\cdot)$. ▮

GGSW [12, 13] propose a WE scheme for $L = L_{\mathrm{ExactCover}}$, one of Karp's original $\mathbf{NP}$-complete problems [17], based on multilinear maps [10]. Its SS[$L_{\mathrm{ExactCover}}$]-security is simply assumed. (It is based on the assumed hardness of a new problem they call Decision Multi-linear No-exact-cover, but this problem is effectively just stating that the WE scheme is SS[$L_{\mathrm{ExactCover}}$]-secure.) We can correspondingly assume their scheme is AS[$L_{\mathrm{ExactCover}}$]-secure, stating a corresponding extension of their assumption.

Below, we'll show that if we are given $\mathsf{WE}_1 \in \mathsf{AS}[L_1]$ for some $\mathbf{NP}$-complete language L_1, then we can transform WE_1 to $\mathsf{WE}_2 \in \mathsf{AS}[L_2]$, for any $L_2 \in \mathbf{NP}$. This was implicit in GGSW for SS. Therefore, it suffices to pick an $\mathbf{NP}$-complete language L_1 (as above) and construct a witness encryption for L_1.

The iO-based construction has perfect correctness. The GGSW construction has perfect correctness under the "dream" version of multi-linear maps, but it is not clear it does if one instantiates the multilinear

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

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

Figure 4: 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.

map with GGH's candidate [10]. See Appendix A for discussion of why relaxing perfect correctness is subtle and needs to be done carefully.

WE for any NP language from WE for an NPC language. Let $\mathsf{R}_1, \mathsf{R}_2$ be **NP**-relations such that there is a Levin reduction (see Section 2) (g, μ, ν) from $\mathcal{L}(\mathsf{R}_2)$ to $\mathcal{L}(\mathsf{R}_1)$. The transform $\mathrm{Trans}_{g,\mu}$ in Fig. 4 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 3.4 below is implicit in [12].

Proposition 3.4 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. 4 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 $m_0, m_1 \in \mathsf{WE}.\mathsf{Msg}$ such that $|m_0| = |m_1|$. Note that $g(x) \in \{0,1\}^* \backslash \mathcal{L}(\mathsf{R}_1)$. Then $\mathsf{Adv}^{\mathsf{ss}}_{\mathsf{WE}_2, \mathcal{L}(\mathsf{R}_2), x, m_0, m_1, A}(\lambda) = \mathsf{Adv}^{\mathsf{ss}}_{\mathsf{WE}_1, \mathcal{L}(\mathsf{R}_1), g(x), m_0, m_1, A}(\lambda)$ 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.

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

$\underline{B(\mathrm{St}, c)}$
$(x, c') \leftarrow c$ ; $b' \leftarrow\!\!\$\ A(\mathrm{St}, c')$
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 $\mathsf{WE}_2 \in \mathsf{AS}[\mathcal{L}(\mathsf{R}_2)]$. ▌

4 Insufficiency of Soundness Security

GGSW [12, 13] 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 [12, 13]. 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.

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. 5. We claim that $\mathsf{SS}[L_G]$-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

```
PKE.Kg($1^\lambda$)
    $sk \leftarrow\$ \{0,1\}^\lambda$ ; $x \leftarrow G(sk)$
    $pk \leftarrow (\lambda, x)$ ; Return $(pk, sk)$

PKE.Enc($pk, m$)
    $(\lambda, x) \leftarrow pk$
    Return $\overline{\mathsf{WE}}$.Enc($1^\lambda, x, m$)

PKE.Dec($sk, c$)
    Return $\overline{\mathsf{WE}}$.Dec($c, sk$)
```

Figure 5: 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 $L_G = \{\, G(s) \,:\, s \in \{0,1\}^* \,\}$.

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 $\mathrm{AS}[L_G]$-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 follows the template of the proof of GGSW [12, 13]. 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 [12, 13] (incorrectly) claim that the result follows from the $\mathrm{SS}[L_G]$-security of $\overline{\mathsf{WE}}$. We instead use the $\mathrm{AS}[L_G]$-security of $\overline{\mathsf{WE}}$, providing a reduction with an explicit construction of an AS adversary.

Proof: Let A be a PT attacking $\mathsf{PKE}[G, \mathsf{WE}]$. Consider the following adversaries B and D:

```
$B(1^\lambda, x)$
    $pk \leftarrow (\lambda, x)$ ; $b \leftarrow\$ \{0,1\}$
    $(m_0, m_1, \mathrm{St}) \leftarrow\$ A(1^\lambda, pk)$
    $c \leftarrow\$ \mathsf{WE.Enc}(1^\lambda, x, m_b)$ ; $b' \leftarrow\$ A(1^\lambda, \mathrm{St}, c)$
    If $b = b'$ then return 1 else return 0

$D(1^\lambda)$
    $x \leftarrow\$ \{0,1\}^{2\lambda}$ ; $pk \leftarrow (\lambda, x)$
    $(m_0, m_1, \mathrm{St}) \leftarrow\$ A(1^\lambda, pk)$ ; $t \leftarrow (\lambda, \mathrm{St})$
    Return $(x, m_0, m_1, t)$

$D(t, c)$
    $(\lambda, \mathrm{St}) \leftarrow t$ ; $b' \leftarrow\$ A(1^\lambda, \mathrm{St}, c)$ ; Return $b'$
```

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

```
MAIN $H_1^A(\lambda)$
    $s \leftarrow\$ \{0,1\}^\lambda$ ; $x \leftarrow G(s)$ ; $b \leftarrow\$ \{0,1\}$ ; passed $\leftarrow$ true
    $pk \leftarrow (\lambda, x)$ ; $(m_0, m_1, \mathrm{St}) \leftarrow\$ A(1^\lambda, pk)$
    $c \leftarrow\$ \mathsf{WE.Enc}(1^\lambda, x, m_b)$ ; $b' \leftarrow\$ A(1^\lambda, \mathrm{St}, c)$
    Return $(b = b') \wedge$ passed

MAIN $H_2^A(\lambda)$, $\boxed{H_3^A(\lambda)}$
    $x \leftarrow\$ \{0,1\}^{2\lambda}$ ; $b \leftarrow\$ \{0,1\}$ ; passed $\leftarrow$ true
    If $x \in L_G$ then bad $\leftarrow$ true ; $\boxed{\text{passed} \leftarrow \text{false}}$
    $pk \leftarrow (\lambda, x)$ ; $(m_0, m_1, \mathrm{St}) \leftarrow\$ A(1^\lambda, pk)$
    $c \leftarrow\$ \mathsf{WE.Enc}(1^\lambda, x, m_b)$ ; $b' \leftarrow\$ A(1^\lambda, \mathrm{St}, c)$
    Return $(b = b') \wedge$ passed
```

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 [6],

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

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

Discussion. 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 $\mathbf{NP}$-complete language L, transform it to $\overline{\mathsf{WE}} \in \mathsf{SS}[L_G]$ via a Levin reduction of L_G to L, 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 B, 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 $\mathbf{NP}$-complete language L via a Levin reduction of L_G to L, 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 $|X(\lambda)| \ge C\lg(\lambda)$ for all $\lambda \in \mathbb{N}$, and therefore we can let $f(\lambda) = \lfloor C\lg(\lambda)\rfloor$.

5 Asymmetric Password-based Encryption

We define asymmetric password-based encryption (A-PBE) and then present a non-intrusive solution based on AS-secure WE.

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

A-PBE security. We view an adversary A as a pair of PT algorithms (A_1, A_2). Adversary $A_1(1^\lambda)$ generates a vector of passwords $\mathbf{pw}$, each entry a $\mathsf{P.pl}(\lambda)$-bit string. 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 has *high min-entropy* if $\mathrm{Guess}_A(\cdot)$ is a negligible function. Note that passwords 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 scheme P is secure if $\mathsf{Adv}^{\text{apbe}}_{\mathsf{P},A}(\lambda) = 2\Pr[\mathrm{APBE}^A_{\mathsf{P}}(\lambda)] - 1$ is negligible for every PT adversary A of high min-entropy, where game $\mathrm{APBE}^A_{\mathsf{P}}(\lambda)$ is defined in Fig. 6. In this game, $A_1(1^\lambda)$ generates a vector of passwords $\mathbf{pw}$, and 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 then makes several oracle queries of the form (m_0, m_1, i) to get $\mathsf{P.Enc}(\mathbf{pw}[i], \mathbf{sa}[i], m_b)$, where m_0, m_1 are equal-length, distinct messages. Finally it outputs a prediction b' for b. The game returns true if the prediction is correct, meaning $b = b'$.

Invasive A-PBE. Let PKE be a PKE scheme and define A-PBE scheme P as follows. $\mathsf{P.Kd}(1^\lambda, sa, pw)$ applies a random oracle RO to $1^\lambda \| sa \| pw$ to get a string r which it uses as coins to compute $(pk, sk) \leftarrow \mathsf{PKE.Kg}(1^\lambda; r)$. It then returns $hpw = pk$ as the "hashed password." $\mathsf{P.Enc}(1^\lambda, pk, sa, m)$ returns $(1^\lambda, sa, \mathsf{PKE.Enc}(pk, m))$. $\mathsf{P.Dec}((1^\lambda, sa, y), pw)$ re-applies RO to $1^\lambda \| sa \| pw$ to get r, re-computes $(pk, sk) \leftarrow \mathsf{PKE.Kg}(1^\lambda; r)$ and returns $m \leftarrow \mathsf{PKE.Dec}(y, sk)$. This A-PBE scheme can be shown to meet our notion of security defined above in the ROM assuming PKE is IND-CPA secure. However, this trivial solution is "invasive" because it prescribes a very particular and non-standard $\mathsf{P.Kd}$. From a practical perspective,

```
Main $\mathrm{APBE}_{\mathsf{P}}^{A}(\lambda)$
  $\mathbf{pw} \leftarrow_{\$} A_1(1^\lambda)$ ; $b \leftarrow_{\$} \{0,1\}$
  For $i = 1$ to $|\mathbf{pw}|$ do
    $\mathbf{sa}[i] \leftarrow_{\$} \{0,1\}^{\mathsf{P.sl}(\lambda)}$ ; $\mathbf{hpw}[i] \leftarrow \mathsf{P.Kd}(1^\lambda, \mathbf{sa}[i], \mathbf{pw}[i])$
  $b' \leftarrow_{\$} A_2^{\mathrm{LR}}(1^\lambda, \mathbf{sa}, \mathbf{hpw})$ ; Return $(b = b')$

LR$(m_0, m_1, i)$
  $c \leftarrow_{\$} \mathsf{P.Enc}(1^\lambda, \mathbf{hpw}[i], \mathbf{sa}[i], m_b)$ ; Return $c$
```

```
Main $\mathrm{KDFR}_{\mathsf{H}}^{A}(\lambda)$
  $\mathbf{pw} \leftarrow_{\$} A_1(1^\lambda)$ ; $b \leftarrow_{\$} \{0,1\}$
  For $i = 1$ to $|\mathbf{pw}|$ do
    $\mathbf{sa}[i] \leftarrow_{\$} \{0,1\}^{\mathsf{H.kl}(\lambda)}$ ; $\mathbf{hpw}[i] \leftarrow \mathsf{H}(1^\lambda, \mathbf{sa}[i], \mathbf{pw}[i])$
  $b' \leftarrow_{\$} A_2(1^\lambda, \mathbf{sa}, \mathbf{hpw})$ ; Return $(b = b')$
```

Figure 6: **Left:** Game APBE defining security of an A-PBE scheme P. **Right:** Game KDFR defining KDF-pseudorandomness for a hash family H.

```
$\mathsf{P}[\mathsf{H},\mathsf{WE}].\mathsf{Kd}(1^\lambda, sa, pw)$
  $hpw \leftarrow \mathsf{H}(1^\lambda, sa, pw)$
  Return $hpw$

$\mathsf{P}[\mathsf{H},\mathsf{WE}].\mathsf{Enc}(1^\lambda, hpw, sa, m)$
  $x \leftarrow (1^\lambda, sa, hpw)$ ; $c \leftarrow_{\$} \mathsf{WE}(1^\lambda, x, m)$
  Return $c$

$\mathsf{P}[\mathsf{H},\mathsf{WE}].\mathsf{Dec}(c, pw)$
  $m \leftarrow \mathsf{WE.Dec}(c, pw)$
  Return $m$
```

Figure 7: A-PBE scheme $\mathsf{P}[\mathsf{H},\mathsf{WE}]$ associated to hash family H and witness encryption scheme WE for L_{H}.

as we noted in Section 1, this is a non-solution, because in-use password hashes are not obtained in this way and the invasive solution will not enable us to provide security with existing, legacy passwords. The real (and technically more interesting) problem is non-invasive A-PBE, where we take $\mathsf{P.Kd}$ 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 $\mathsf{P.Kd}$ function of PKCS#5 or other standards.

Non-invasive A-PBE. We aim to design a non-invasive A-PKE scheme P such that $\mathsf{P.Kd}$ follows existing standards like PKCS#5 [18]. So we demand that $\mathsf{P.Kd}$ be a keyed hash function family H. Here $\mathsf{H}(1^\lambda, \cdot, \cdot) : \{0,1\}^{\mathsf{H.kl}(\lambda)} \times \{0,1\}^{\mathsf{H.il}(\lambda)} \to \{0,1\}^{\mathsf{H.ol}(\lambda)}$ for every $\lambda \in \mathbb{N}$, where $\mathsf{H.kl}, \mathsf{H.il}, \mathsf{H.ol} : \mathbb{N} \to \mathbb{N}$ are the key-length function, input-length function, and output-length function of H respectively. 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 $\mathsf{P}[\mathsf{H},\mathsf{WE}]$ specified in Fig. 7. We let $\mathsf{P}[\mathsf{H},\mathsf{WE}].\mathsf{Msg} = \mathsf{WE.Msg}, \mathsf{P}[\mathsf{H},\mathsf{WE}].\mathsf{pl} = \mathsf{H.il}$, $\mathsf{P}[\mathsf{H},\mathsf{WE}].\mathsf{sl} = \mathsf{H.kl}$ and $\mathsf{P}[\mathsf{H},\mathsf{WE}].\mathsf{hl} = \mathsf{H.ol}$. The construction lets the salt play the role of the key for H, the password being the input and the hashed password the output.

KDF pseudorandomness. We now formalize a hardness assumption on the family H. We say that H is KDF-pseudorandom if $\mathsf{Adv}^{\mathsf{kdfr}}_{\mathsf{H},A}(\lambda) = 2[\mathrm{KDFR}^{A}_{\mathsf{H}}(\lambda)] - 1$ is negligible for any PT adversary $A = (A_1, A_2)$ of high min-entropy, where game $\mathrm{KDFR}^{A}_{\mathsf{H}}$ is shown at the right panel of Fig. 6. 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.

Results. The following says that if H is KDF-pseudorandom and WE is $\mathrm{AS}[L_{\mathsf{H}}]$-secure then $\mathsf{P}[\mathsf{H},\mathsf{WE}]$ is a secure A-PBE scheme.

Theorem 5.1 Let H be a hash function family such that $2^{\mathsf{H.il}(\cdot)-\mathsf{H.ol}(\cdot)}$ is a negligible function. If H is KDF-pseudorandom and $\mathsf{WE} \in \mathsf{AS}[L_{\mathsf{H}}]$ then $\mathsf{P}[\mathsf{H},\mathsf{WE}]$ 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.

Proof of Theorem 5.1: Let $A = (A_1, A_2)$ be a PT adversary of high min-entropy attacking $\mathsf{P}[\mathsf{H},\mathsf{WE}]$. Let $B = (B_1, B_2)$ be an adversary attacking H as follows. Since B_1 is exactly A_1, and A is of high min-entropy, B also has high min-entropy.

```
$\underline{B_1(1^\lambda)}$
$\mathbf{pw} \leftarrow\!\!\$\; A_1(1^\lambda)$ ; Return $\mathbf{pw}$
$\underline{B_2(1^\lambda, \mathbf{sa}, \mathbf{hpw})}$
$b \leftarrow\!\!\$\; \{0,1\}$ ; $b' \leftarrow\!\!\$\; A_2^{\text{LRSim}}(1^\lambda, \mathbf{sa}, \mathbf{hpw})$
If $(b = b')$ then return 1 else return 0
```

```
$\underline{\text{LRSim}(m_0, m_1, i)}$
$x \leftarrow (1^\lambda, \mathbf{sa}[i], \mathbf{hpw}[i])$ ; $c \leftarrow\!\!\$\; \mathsf{WE.Enc}(1^\lambda, x, m_b)$
Return $c$
```

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 $i \le |\mathbf{hpw}|$. Recall that A may make several oracle queries but D is allowed only a single query (x, m_0, m_1, 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 A produces its own query (x, m_0, m_1, 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 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
  $\mathbf{sa}[i] \leftarrow\!\!\$\; \{0,1\}^{\mathsf{H.kl}(\lambda)}$ ; $\mathbf{hpw}[i] \leftarrow\!\!\$\; \{0,1\}^{\mathsf{H.ol}(\lambda)}$
$j \leftarrow 1$ ; $s \leftarrow\!\!\$\; \{1, \ldots, q(\lambda)\}$ ; $A_2^{\text{LRSim}}(1^\lambda, \mathbf{sa}, \mathbf{hpw}; r)$
$\text{St} \leftarrow (1^\lambda, \mathbf{c}, s, r, \mathbf{sa}, \mathbf{hpw})$
$(x, m_0, m_1) \leftarrow p$ ; Return $(x, m_0, m_1, \text{St})$
$\underline{D(\text{St}, c)}$
$(1^\lambda, \mathbf{c}, s, r, \mathbf{sa}, \mathbf{hpw}) \leftarrow \text{St}$
$\mathbf{c}[s] \leftarrow c$ ; $b' \leftarrow A_2^{\text{LRSim}}(1^\lambda, \mathbf{sa}, \mathbf{hpw}; r)$ ; Return $b'$
```

```
$\underline{\text{LRSim}(m_0, m_1, i)}$
$x \leftarrow (1^\lambda, \mathbf{sa}[i], \mathbf{hpw}[i])$
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 $\text{St} = \bot$ then $\mathbf{c}[j] \leftarrow c$ else $c \leftarrow \mathbf{c}[j]$
$j \leftarrow j + 1$ ; Return $c$
```

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

```
$\underline{\text{Main } H_1^A(\lambda), \boxed{H_2^A(\lambda)}}$
$\mathbf{pw} \leftarrow\!\!\$\; A_1(1^\lambda)$ ; $b \leftarrow\!\!\$\; \{0,1\}$
For $i = 1$ to $|\mathbf{pw}|$ do
  $\mathbf{sa}[i] \leftarrow\!\!\$\; \{0,1\}^{\mathsf{H.kl}(\lambda)}$
  $\mathbf{hpw}[i] \leftarrow \mathsf{H}(1^\lambda, \mathbf{sa}[i], \mathbf{pw}[i])$ ; $\boxed{\mathbf{hpw}[i] \leftarrow\!\!\$\; \{0,1\}^{\mathsf{H.ol}(\lambda)}}$
$b' \leftarrow\!\!\$\; A_2^{\text{LR}}(1^\lambda, \mathbf{sa}, \mathbf{hpw})$
Return $(b = b')$
```

```
$\underline{\text{LR}(m_0, m_1, i)}$
$x \leftarrow (1^\lambda, \mathbf{sa}[i], \mathbf{hpw}[i])$
Return $\mathsf{WE.Enc}(1^\lambda, x, m_b)$
```

On the one hand,

$$\begin{aligned} \Pr[\,\text{KDFR}_{\mathsf{H}}^B(\lambda) \Rightarrow \mathsf{true} \,|\, a = 1\,] &= \Pr[H_1^A(\lambda)] = \Pr[\text{APBE}_{\mathsf{P[H,WE]}}^A(\lambda)], \text{ and} \\ \Pr[\,\text{KDFR}_{\mathsf{H}}^B(\lambda) \Rightarrow \mathsf{false} \,|\, a = 0\,] &= \Pr[H_2^A(\lambda)] \end{aligned}$$

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

$$2\Pr[\text{AS}_{\mathsf{WE},L_{\mathsf{H}}}^D(\lambda)] - 1 \ge \frac{1}{q(\lambda)}(\Pr[\,H_2^A(\lambda) \Rightarrow \mathsf{true} \,|\, d = 1\,] - \Pr[\,H_2^A(\lambda) \Rightarrow \mathsf{false} \,|\, d = 0\,]) - 2^{\mathsf{H.il}(\lambda)-\mathsf{H.ol}(\lambda)+1},$$

for every $\lambda \in \mathbb{N}$, where d is the challenge bit b that game H_2^A samples. Summing up, $\mathsf{Adv}_{\mathsf{P[H,WE]},A}^{\mathsf{apbe}}(\lambda) \le 2\mathsf{Adv}_{\mathsf{H},B}^{\mathsf{kdfr}}(\lambda) + q(\lambda) \cdot \mathsf{Adv}_{\mathsf{WE},L_{\mathsf{H}},D}^{\mathsf{as}}(\lambda) + q(\lambda) \cdot 2^{\mathsf{H.il}(\lambda)-\mathsf{H.ol}(\lambda)+1}$, for every $\lambda \in \mathbb{N}$, and thus $\mathsf{P[H,WE]}$ 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.

```
MAIN $G_s^A(\lambda)$, $\boxed{P_s^A(\lambda)}$
$\mathbf{pw} \leftarrow_\$ A_1(1^\lambda)$ ; $b \leftarrow_\$ \{0,1\}$ ; $\mathsf{passed} \leftarrow \mathsf{true}$
For $i = 1$ to $|\mathbf{pw}|$ do
  $\mathbf{sa}[i] \leftarrow_\$ \{0,1\}^{\mathsf{H.kl}(\lambda)}$ ; $\mathbf{hpw}[i] \leftarrow_\$ \{0,1\}^{\mathsf{H.ol}(\lambda)}$
$j \leftarrow 1$ ; $b' \leftarrow_\$ A_2^{\mathrm{LR}}(1^\lambda, \mathbf{sa}, \mathbf{hpw})$
Return $(b = b') \wedge \mathsf{passed}$
```

```
$\mathrm{LR}(m_0, m_1, i)$
$x \leftarrow (1^\lambda, \mathbf{sa}[i], \mathbf{hpw}[i])$ ; $m \leftarrow m_b$
If $j = s$ then
  If $x \in L_{\mathsf{H}}$ then $\mathsf{bad} \leftarrow \mathsf{true}$ ; $\boxed{\mathsf{passed} \leftarrow \mathsf{false}}$
If $j < s$ then $m \leftarrow m_0$ elsif $j > s$ then $m \leftarrow m_1$
$j \leftarrow j + 1$ ; Return $\mathsf{WE.Enc}(1^\lambda, x, m)$
```

For each $s \in \{1, \ldots, q(\lambda)\}$, games G_s^A and P_s^A are identical-until-bad, and from the fundamental lemma of game-playing [6],

$$\Pr[G_s^A(\lambda)] - \Pr[P_s^A(\lambda)] \le \Pr[G_s^A(\lambda) \text{ sets } \mathsf{bad}] \le 2^{\mathsf{H.il}(\lambda) - \mathsf{H.ol}(\lambda)}$$

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_s^A samples. Then $\Pr[\, G_s^A(\lambda) \Rightarrow \mathsf{true} \,|\, b_s = 1\,] = \Pr[\, G_{s-1}^A(\lambda) \Rightarrow \mathsf{false} \,|\, b_{s-1} = 0\,]$ for every $s \in \{2, 3, \ldots, q(\lambda)\}$ and every $\lambda \in \mathbb{N}$, and thus

$$\begin{aligned}
\sum_{s=1}^{q(\lambda)} \bigl(2\Pr[G_s^A(\lambda)] - 1\bigr) &= \sum_{s=1}^{q(\lambda)} \bigl(\Pr[\, G_s^A(\lambda) \Rightarrow \mathsf{true} \,|\, b_s = 1\,] - \Pr[\, G_s^A(\lambda) \Rightarrow \mathsf{false} \,|\, b_s = 0\,]\bigr) \\
&= \Pr[\, G_1^A(\lambda) \Rightarrow \mathsf{true} \,|\, b_1 = 1\,] - \Pr[\, G_{q(\lambda)}^A(\lambda) \Rightarrow \mathsf{false} \,|\, b_{q(\lambda)} = 0\,] \\
&= \Pr[\, H_2^A(\lambda) \Rightarrow \mathsf{true} \,|\, d = 1\,] - \Pr[\, H_2^A(\lambda) \Rightarrow \mathsf{false} \,|\, d = 0\,] \qquad (2)
\end{aligned}$$

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

$$\begin{aligned}
-1 + 2 \cdot \Pr[\mathrm{AS}_{\mathsf{WE}, L_{\mathsf{H}}}^D(\lambda)] &= -1 + \frac{2}{q(\lambda)} \sum_{s=1}^{q(\lambda)} \Pr[P_s^A(\lambda)] \\
&\ge -1 + \frac{2}{q(\lambda)} \sum_{s=1}^{q(\lambda)} \bigl(\Pr[G_s^A(\lambda)] - 2^{\mathsf{H.il}(\lambda) - \mathsf{H.ol}(\lambda)}\bigr) \\
&= -2^{\mathsf{H.il}(\lambda) - \mathsf{H.ol}(\lambda) + 1} + \frac{1}{q(\lambda)} \sum_{s=1}^{q(\lambda)} \bigl(2\Pr[G_s^A(\lambda)] - 1\bigr) \qquad (3)
\end{aligned}$$

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

DISCUSSION. In the result above, we require that $2^{\mathsf{H.il}(\cdot) - \mathsf{H.ol}(\cdot)}$ be a negligible function, that is, 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 78-bit) and H is iterated SHA-1 (output length is 160-bit). However, when passwords are longer, say 24-character, then Theorem 5.1 doesn't apply. This is unsatisfying, because password length is a measure of password strength, so intuitively, longer passwords should offer better security. In Section 6, we formalize a stronger security requirement for witness encryption that allows us to remove the assumption on the input/output length of H.

6 A-PBE from Extractable 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 [15, 14] 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 [15, 14] in being adaptive in the vein of AS, but weakens it by not involving auxiliary inputs. The formalizations also differ in other details.

```
$\underline{\textsc{Main}\ \mathrm{XS}^{A,E}_{\mathsf{WE},\mathsf{R}}(\lambda)}$
$(x, m_0, m_1, \mathrm{St}) \leftarrow\!\!\$\ A(1^\lambda)\ ;\ b \leftarrow\!\!\$\ \{0,1\}$
$c \leftarrow\!\!\$\ \mathsf{WE.Enc}(1^\lambda, x, m_b)\ ;\ b' \leftarrow\!\!\$\ A(\mathrm{St}, c)$
$w \leftarrow\!\!\$\ E(1^\lambda, x, m_0, m_1, \mathrm{St}, c)$
Return $((b = b') \wedge \neg\mathsf{R}(x, w))$
```

```
$\underline{\textsc{Main}\ \mathrm{KDFO}^{A}_{\mathsf{H}}(\lambda)}$
$\mathbf{pw} \leftarrow\!\!\$\ A_1(1^\lambda)$
For $i = 1$ to $|\mathbf{pw}|$ do
    $\mathbf{sa}[i] \leftarrow\!\!\$\ \{0,1\}^{\mathsf{H.kl}(\lambda)}\ ;\ \mathbf{hpw}[i] \leftarrow \mathsf{H}(1^\lambda, \mathbf{sa}[i], \mathbf{pw}[i])$
$(w, i) \leftarrow\!\!\$\ A_2(1^\lambda, \mathbf{sa}, \mathbf{hpw})$
Return $(\mathbf{hpw}[i] = \mathsf{H}(1^\lambda, \mathbf{sa}[i], w))$
```

Figure 8: **Left:** Game XS defining extractable security of witness encryption scheme WE. **Right:** Game KDFO defining KDF one-wayness of H.

XS-SECURE WITNESS ENCRYPTION. Let R be an $\mathbf{NP}$-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 $\mathsf{Adv}^{\mathsf{xs}}_{\mathsf{WE},\mathsf{R},A,E}(\lambda) = 2\Pr[\mathrm{XS}^{A,E}_{\mathsf{WE},\mathsf{R}}(\lambda)] - 1$ is negligible, where game $\mathrm{XS}^{A,E}_{\mathsf{WE},\mathsf{R}}$ is defined at the left panel of Fig. 8. Let $\mathsf{XS}[L]$ denote the set of correct, XS[L]-secure witness encryption schemes for L.

RELATION WITH AS SECURITY. 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 6.1 below formally confirms this.

Proposition 6.1 For any $\mathbf{NP}$-relation R, it holds that $\mathsf{XS}[\mathcal{L}(\mathsf{R})] \subseteq \mathsf{AS}[\mathcal{L}(\mathsf{R})]$.

Proof: 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 $\mathsf{Adv}^{\mathsf{xs}}_{\mathsf{WE},\mathsf{R},A,E}(\cdot)$ is negligible. Consider the following games H_1 and H_2; the latter includes the boxed statement but the former does not.

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

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

ACHIEVING XS SECURITY. Boyle, Chung, and Pass [9] introduce the notion of *extractability obfuscation* (xO), and show that it implies extractable witness encryption meeting GKPVZ's definition [15]. We now give an alternative definition of xO and show that it implies XS[$\mathcal{L}(\mathsf{R})$]-secure witness encryption, for any $\mathbf{NP}$ relation R. Let F be an obfuscator, defining a PT obfuscation algorithm $\mathsf{F.Ob}$ and a PT evaluation algorithm $\mathsf{F.Ev}$. We say that F is xO-secure if for every PT adversary A, there is a PT algorithm (extractor) E such that $\mathsf{Adv}^{\mathsf{xo}}_{\mathsf{F},A,E}(\lambda) = 2\Pr[\mathrm{XO}^{A,E}_{\mathsf{F}}(\lambda)] - 1$ is negligible, where game XO is defined at as follows:

```
$\underline{\textsc{Main}\ \mathrm{XO}^{A,E}_{\mathsf{F}}(\lambda)}$
$(C_0, C_1, \mathrm{St}) \leftarrow\!\!\$\ A(1^\lambda)\ ;\ b \leftarrow\!\!\$\ \{0,1\}\ ;\ c \leftarrow\!\!\$\ \mathsf{F.Ob}(1^\lambda, C_b)$
$b' \leftarrow\!\!\$\ A(\mathrm{St}, c)\ ;\ w \leftarrow\!\!\$\ E(1^\lambda, C_0, C_1, \mathrm{St}, c)$
Return $(b = b') \wedge (C_0(w) = C_1(w))$
```

In the game above, circuits C_0, C_1 must have the same size. Recall that in Section 3, we have the construction $\mathsf{WE}_{\mathsf{R}}[\mathsf{F}]$ of witness encryption for language $\mathcal{L}(\mathsf{R}) \in \mathbf{NP}$ from obfuscator F. The following says that if F is assumed to be xO-secure then $\mathsf{WE}_{\mathsf{R}}[\mathsf{F}]$ is XS[$\mathcal{L}(\mathsf{R})$]-secure.

Theorem 6.2 Let R be an **NP** relation, and let F be an obfuscator. Construct $\mathsf{WE_R[F]}$ as in Section 3. If F is xO-secure then $\mathsf{WE_R[F]} \in \mathsf{XS}[\mathcal{L}(\mathsf{R})]$.

Proof: For each $x, m \in \{0,1\}^*$, let $R_{x,m}$ be a circuit that outputs 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 $\mathsf{WE_R[F]}$. Since A needs to produce distinct messages m_0, m_1, and $\mathsf{WE_R[F].Msg} = \{0,1\}$, wlog, assume that $m_0 = 0$ and $m_1 = 1$. Consider the following adversary B attacking F.

```
$\underline{B(1^\lambda)}$
$(x, m_0, m_1, \mathrm{St}) \leftarrow\!\!\$\; A(1^\lambda)$ ; Return $(R_{x,0}, R_{x,1}, \mathrm{St})$

$\underline{B(\mathrm{St}, c)}$
$b' \leftarrow\!\!\$\; A(\mathrm{St}, c)$ ; Return $b'$
```

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

```
$\underline{\overline{E}(1^\lambda, x, m_0, m_1, \mathrm{St}, c)}$
$w \leftarrow\!\!\$\; E(1^\lambda, R_{x,0}, R_{x,1}, \mathrm{St}, c)$ ; Return $w$
```

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

KDF ONE-WAYNESS. We now formalize another hardness assumption, KDF 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 key-derivation functions: if passwords are well-chosen to resist dictionary attacks, then no adversary should be able to recover some password from the derived keys. Formally, we say that H is KDF one-way if $\mathsf{Adv}^{\mathsf{kdfo}}_{\mathsf{H},A}(\lambda) = \Pr[\mathrm{KDFO}^A_{\mathsf{H}}(\lambda]$ is negligible for all PT adversary $A = (A_1, A_2)$ of high min-entropy, where game $\mathrm{KDFO}^A_{\mathsf{H}}$ is shown at the right panel of Fig. 8 and high min-entropy of A was defined in Section 5.

RESULTS. The following establishes the security of $\mathsf{P[H,WE]}$, making no assumption on the input/output length of H.

Theorem 6.3 If H is KDF one-way and $\mathsf{WE} \in \mathsf{XS}[L_{\mathsf{H}}]$ then $\mathsf{P[H,WE]}$ is a secure A-PBE scheme.

Proof: Let $A = (A_1, A_2)$ be a PT adversary of high min-entropy attacking $\mathsf{P[H,WE]}$. 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 A 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
    $\mathbf{sa}[i] \leftarrow_{\$} \{0,1\}^{\mathsf{H.kl}(\lambda)}$ ; $\mathbf{hpw}[i] \leftarrow \mathsf{H}(1^\lambda, \mathbf{sa}[i], \mathbf{pw}[i])$
$j \leftarrow 1$ ; $s \leftarrow_{\$} \{1, \ldots, q(\lambda)\}$ ; $A_2^{\text{LRSim}}(1^\lambda, \mathbf{sa}, \mathbf{hpw}; r)$
$\mathrm{St} \leftarrow (1^\lambda, \mathbf{c}, s, r, \mathbf{sa}, \mathbf{hpw})$
$(x, m_0, m_1) \leftarrow p$ ; Return $(x, m_0, m_1, \mathrm{St})$

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

```
$\underline{\text{LRSim}(m_0, m_1, i)}$
$x \leftarrow (1^\lambda, \mathbf{sa}[i], \mathbf{hpw}[i])$
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 $c$
```

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 of high min-entropy, B also has high min-entropy.

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

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

```
$\underline{\text{LRSim}(m_0, m_1, i)}$
$x \leftarrow (1^\lambda, \mathbf{sa}[i], \mathbf{hpw}[i])$ ; $m \leftarrow m_b$
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 $s \in \{1, \ldots, q(\lambda)\}$.

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

```
$\underline{\text{LR}(m_0, m_1, i)}$
$x \leftarrow (1^\lambda, \mathbf{sa}[i], \mathbf{hpw}[i])$ ; $m \leftarrow m_b$
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

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

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

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

where d is the challenge bit of game $\mathrm{APBE}_{\mathsf{P[H,WE]}}$. On the other hand,

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

Hence, summing up, $\mathsf{Adv}^{\mathsf{apbe}}_{\mathsf{P[H,WE]},A}(\cdot) \leq 2q\cdot\mathsf{Adv}^{\mathsf{kdfo}}_{\mathsf{H},B}(\cdot)+q\cdot\mathsf{Adv}^{\mathsf{xs}}_{\mathsf{WE},\mathsf{R_H},D,E}(\cdot)$, and thus $\mathsf{P[H,WE]}$ is a secure A-PBE scheme. ▌

Acknowledgments

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

References

[1] 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, 9

[2] 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. 6

[3] M. Bellare, S. Meiklejohn, and S. Thomson. Key-versatile signatures and applications: Rka, kdm and joint enc/sig. Cryptology ePrint Archive, Report 2013/326, 2013. 5

[4] 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. 4

[5] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93*, pages 62–73. ACM Press, Nov. 1993. 3

[6] 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. 6, 11, 15

[7] N. Bitansky, R. Canetti, O. Paneth, and A. Rosen. Indistinguishability obfuscation vs. auxiliary-input extractable functions: One must fall. Cryptology ePrint Archive, Report 2013/641, 2013. 5

[8] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudorandom bits. *SIAM Journal on Computing*, 13(4):850–864, 1984. 6

[9] E. Boyle, K.-M. Chung, and R. Pass. On extractability obfuscation. Cryptology ePrint Archive, Report 2013/650, 2013. 5, 16

[10] S. Garg, C. Gentry, and S. Halevi. Candidate multilinear maps from ideal lattices. In *Advances in Cryptology–EUROCRYPT 2013*, pages 1–17, 2013. 4, 5, 9, 10, 21

[11] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. *Foundations of Computer Science (FOCS)*, 2013. 4, 6, 9

[12] S. Garg, C. Gentry, A. Sahai, and B. Waters. Witness encryption and its applications. In *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing*, pages 467–476. ACM, 2013. 3, 4, 5, 7, 9, 10, 11, 20

[13] S. Garg, C. Gentry, A. Sahai, and B. Waters. Witness encryption and its applications. Cryptology ePrint Archive, Report 2013/258, version 20130508:202916, 2013. 3, 4, 5, 7, 9, 10, 11, 20

[14] S. Goldwasser, Y. Kalai, R. A. Popa, V. Vaikuntanathan, , and N. Zeldovich. How to run turing machines on encrypted data. Cryptology ePrint Archive, Report 2013/229, 2013. 5, 15

[15] S. Goldwasser, Y. Kalai, R. A. Popa, V. Vaikuntanathan, and N. Zeldovich. How to run turing machines on encrypted data. In *Advances in Cryptology - CRYPTO 2013*, pages 536–553, 2013. 5, 15, 16

[16] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984. 6

[17] R. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103, 1972. 9

[18] PKCS #5: Password-based cryptography standard (rfc 2898). RSA Data Security, Inc., Sept. 2000. Version 2.0. 3, 4, 13

[19] T. Ristenpart, H. Shacham, and T. Shrimpton. Careful with composition: Limitations of the indifferentiability framework. In K. G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 487–506. Springer, May 2011. 6

[20] A. Sahai. Personal Communication, October 2013. 20, 21

[21] A. C. Yao. Theory and applications of trapdoor functions. In *23rd FOCS*, pages 80–91. IEEE Computer Society Press, Nov. 1982. 6

A Relaxation of correctness

Let R be an **NP** relation and let WE be a witness encryption for $L = \mathcal{L}(\mathsf{R})$. GGSW [12, 13] define a relaxed notion of correctness so that $\mathsf{WE.Dec}$ may sometimes fail. Their definition is quoted below, with the notation adjusted:

> For any security parameter λ, for any $m \in \mathsf{WE.Msg}$, and for any $x \in L$ such that $R(x, w)$ holds, we have that $\Pr[\mathsf{WE.Dec}(\mathsf{WE.Enc}(1^\lambda, x, m), w) = m] = 1 - \mathsf{negl}(\lambda)$.

This definition is ambiguous, because the implicit negligible function in the notation $\mathsf{negl}(\cdot)$ is not quantified, and thus it is unclear if it depends on x, w, and m or not. There are several ways to interpret the definition, of which two natural ones are the following:

- **Weak correctness:** For every $x \in \mathcal{L}(\mathsf{R})$, every $w \in \mathsf{R}(x)$ and every $m \in \mathsf{WE.Msg}$ there is a negligible function ν such that for all $\lambda \in \mathbb{N}$ we have that $\Pr[\mathsf{WE.Dec}(\mathsf{WE.Enc}(1^\lambda, x, m), w) = m] \geq 1 - \nu(\lambda)$.
- **Uniform correctness:** There is a negligible function ν such that, for every $\lambda \in \mathbb{N}$, every $x \in \mathcal{L}(\mathsf{R})$, every $w \in \mathsf{R}(x)$ and every $m \in \mathsf{WE.Msg}$, we have that $\Pr[\mathsf{WE.Dec}(\mathsf{WE.Enc}(1^\lambda, x, m), w) = m] \geq 1 - \nu(\lambda)$.

We stress that in uniform correctness, function ν is independent of x, w, and m. Weak correctness seems to be a more plausible interpretation of what GGSW meant, as it mimics the quantification used in SS security. We communicated this issue to GGSW, saying we felt their definition was ambiguous, giving our candidate interpretations, and asking them which, if any, was what they meant. In response, Sahai [20] remarked that (i) weak correctness may affect usability of witness encryption, as users might not know how to choose the value for λ for a specific x, and (ii) it's unclear if there is a witness encryption scheme that doesn't achieve perfect correctness but still satisfies strong correctness, because the scheme needs to be sensitive to $|x|$.

What we note here is that weak correctness has more serious defects than GGSW appear to be aware of. Let $\mathsf{WAS}[L]$ be the set of all weakly correct and adaptively secure witness encryption schemes for a language $L \in \mathbf{NP}$. We argue that there are WAS-secure witness encryption schemes that make GGSW's applications (PKE, IBE, and ABE for circuits) fail to guarantee any correctness, regardless of how one chooses λ. Consider, for example, the PKE scheme $\mathsf{PKE}[G, \overline{\mathsf{WE}}]$ specified in Fig. 5, where G is a length-doubling secure PRG, and $\overline{\mathsf{WE}}$ is a witness encryption scheme for $L_G = \{ G(s) : s \in \{0,1\}^* \}$. We claim that weak correctness of $\overline{\mathsf{WE}}$ is insufficient for PKE, meaning we give an example of a particular witness encryption scheme $\overline{\mathsf{WE}} \in \mathsf{WAS}[L_G]$ such that the ciphertext produced by $\mathsf{PKE}[G, \overline{\mathsf{WE}}]$ is always $(0,0)$, for any messages and any $\lambda \in \mathbb{N}$. Starting from an arbitrary $\mathsf{WE} \in \mathsf{WAS}[L]$, we can modify it to $\overline{\mathsf{WE}}$ that misbehaves, sending $(0,0)$ for all messages if $|x| \geq 2\lambda$, and otherwise behaves like WE. Scheme $\overline{\mathsf{WE}}$ is formally specified below; we let $\overline{\mathsf{WE}}.\mathsf{Msg} = \mathsf{WE}.\mathsf{Msg}$.

```
$\overline{\mathsf{WE}}.\mathsf{Enc}(1^\lambda, x, m)$
If $|x| \geq 2\lambda$ then return $(0,0)$
Else return $(1, \mathsf{WE}.\mathsf{Enc}(1^\lambda, x, m))$

$\overline{\mathsf{WE}}.\mathsf{Dec}(c, w)$
$(b, t) \leftarrow c$
If $b = 0$ then return $\perp$ else return $\mathsf{WE}.\mathsf{Dec}(t, w)$
```

We claim that $\overline{\mathsf{WE}} \in \mathsf{WAS}[L_G]$. Then, for any message m, scheme $\mathsf{PKE}[G, \overline{\mathsf{WE}}]$ always sends a string x of length 2λ to $\overline{\mathsf{WE}}.\mathsf{Enc}(1^\lambda, \cdot, m)$, and thus the ciphertext will be $(0,0)$. To justify the claim above, note that the adaptive soundness of scheme $\overline{\mathsf{WE}}$ follows from that of scheme WE. For weak correctness, fix $x \in \mathcal{L}(\mathsf{R})$, $w \in \mathsf{R}(x)$, and $m \in \mathsf{WE}.\mathsf{Msg}$. Let ν be the negligible function such that

$$\Pr[\mathsf{WE}.\mathsf{Dec}(\mathsf{WE}.\mathsf{Enc}(1^\lambda, x, m), w) = m] \geq 1 - \nu(\lambda);$$

this function exists because $\mathsf{WE} \in \mathsf{WAS}[L_G]$. Let $\overline{\nu} : \mathbb{N} \to \mathbb{N}$ be the function that $\overline{\nu}(\lambda) = \nu(\lambda)$ if $\lambda > |x|/2$, and $\overline{\nu}(\lambda) = 1$ otherwise. Function $\overline{\nu}$ is also negligible, and

$$\Pr[\overline{\mathsf{WE}}.\mathsf{Dec}(\overline{\mathsf{WE}}.\mathsf{Enc}(1^\lambda, x, m), w) = m] \geq 1 - \overline{\nu}(\lambda)$$

for every $\lambda \in \mathbb{N}$, justifying the weak correctness of $\overline{\mathsf{WE}}$. Hence $\overline{\mathsf{WE}} \in \mathsf{WAS}[L_G]$, as claimed. One can also build similar counter-examples for other applications of GGSW.

We now formalize a notion of correctness that can be used for the applications above; this definition is also independently suggested by Sahai [20].

- **Strong correctness:** For every polynomial p there is a negligible function ν such that for every $\lambda \in \mathbb{N}$, every $x \in \mathcal{L}(\mathsf{R})$, every $w \in \mathsf{R}(x)$ and every $m \in \mathsf{WE}.\mathsf{Msg}$, if $|x|, |m| \leq p(\lambda)$ then
$$\Pr[\mathsf{WE}.\mathsf{Dec}(\mathsf{WE}.\mathsf{Enc}(1^\lambda, x, m), w) = m] \geq 1 - \nu(\lambda) \ .$$

We stress that the function ν is independent of the choice of x, w, and m. When one instantiates GGSW's witness encryption scheme from GGH's candidate for multilinear maps [10], if one sets appropriately large values for the parameters of the lattices in GGH's construction then the corresponding witness encryption scheme will satisfy strong correctness.

B Extending counter-examples for GGSW's PKE scheme

Recall that in Section 4, we have built a counter-example for scheme $\mathsf{PKE}[G, \overline{\mathsf{WE}}]$ (specified in Fig. 5) 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 $\mathbf{NP}$-complete language $L = \mathcal{L}(\mathsf{R})$, transform it to $\overline{\mathsf{WE}} \in \mathsf{SS}[L_G]$ via the transform in Fig. 4 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 $\mathbf{NP}$-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. 4. 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.