Relatively-Sound NIZKs and Password-Based Key-Exchange

Charanjit S. Jutla
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598

Arnab Roy
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598

Abstract

We define a new notion of relatively-sound non-interactive zero-knowledge (NIZK) proofs, where a private verifier with access to a trapdoor continues to be sound even when the Adversary has access to simulated proofs and common reference strings. It is likely that this weaker notion of relative-soundness suffices in most applications that need simulation-soundness. We show that for certain languages which are diverse groups, and hence allow smooth projective hash functions, one can obtain more efficient single-theorem relatively-sound NIZKs as opposed to simulation-sound NIZKs. We also show that such relatively-sound NIZKs can be used to build rather efficient publicly-verifiable CCA2-encryption schemes.

By employing this new publicly-verifiable encryption scheme along with an associated smooth projective-hash, we show that a recent PAK-model single-round password-based key exchange protocol of Katz and Vaikuntanathan, Proc. TCC 2011, can be made much more efficient. We also show a new single round UC-secure password-based key exchange protocol with only a constant number of group elements as communication cost, whereas the previous single round UC-protocol required $\Omega(k)$ group elements, where k is the security parameter.

Keywords: NIZK, simulation-sound, PAKE, UC, smooth hash, pairings, DLIN.

Contents

1 Introduction

Authentication based on passwords is a significant security paradigm in today's world. Security in this scenario has been a challenging problem to solve because passwords typically come from low-entropy domains which results in insufficient randomness to generate cryptographically secure keys. Gong et al [14] raised the problem of designing protocols resistant to offline password guessing attacks. Roughly, the goal of security should be that other than guessing the low-entropy password by an online attack, the protocol must otherwise provide strong security based on a security parameter. Halevi and Krawczyk [17] formalized this definition and gave proofs of security where one of the party has a public key.

The setting in which only passwords are shared by the peers was first considered by Bellovin and Merritt [3]. Formal definitions in this setting were given by [2, 5, 22], now referred to as the PAK-security model. They also proved the Bellovin and Merritt protocol secure in the ideal-cipher model. Goldreich and Lindell [13] introduced a third security definition and also gave the first provably secure protocol in the standard model. A practical and provably secure three round protocol in the common reference string setting was first developed by Katz et al [18], which was subsequently generalized by Gennaro and Lindell [12]. Starting with [18], these protocols employ smooth projective hash functions which have been a standard tool in cryptography ever since Cramer and Shoup defined them to give an efficient chosen ciphertext secure (CCA2) encryption scheme [10].

As illustrated by Gennaro and Lindell [12], who call this the non-malleable commitment paradigm, these protocols require the two peers A and B to non-malleably commit to their password to their peer (say B), e.g. by CCA2-encrypting the password under a public key given as a common reference string (CRS). While, the peer B cannot decrypt this commitment, it might be able to compute a smooth projective-hash on this commitment using a smooth hash key that it generates. The projection of this smooth hash key is sent to peer A, and peer A can compute the same smooth hash using the witness it has for the commitment. The two peers then output a product of two such smooth hashes, one for its own commitment and one for its peer. The problem, however, is that smooth projective-hash for the language, which in this case is the CCA2-ciphertext encrypting a password, is not easy to define, and [12] require an adaptive smooth hash key, which makes the key-exchange protocol a multi-round protocol.

Recently, Katz and Vaikuntanathan [19] gave a single round protocol for password-based authenticated key exchange, by utilizing a publicly-verifiable CCA2-encryption scheme of Sahai [26]. A publicly-verifiable encryption scheme allows a (non-interactive) public verification of well-formedness of the ciphertext, i.e. it returns TRUE if and only if the decryption oracle will not return an 'invalid ciphertext' response when queried with this ciphertext[1]. The public verification allows the smooth hash to be defined on only a projection of the ciphertext, which in their case happens to be two El-Gamal encryptions of the password. Such smooth hashes are easy to define and compute.

While the resulting protocol requires only a constant number of group elements, as it employs

[1] [21] describe a publicly verifiable CCA scheme under a dynamic assumption q-BDHI, whereas in this paper we are interested in proving results under static assumptions, e.g. SXDH, DLIN. More general verifiable encryption schemes have been considered in [6] which verify additional properties of the underlying message, but that verification is only non-interactive in the Random Oracle model.

simulation-sound Groth-Sahai NIZKs [16], under the decisional-linear assumption (DLIN [4]) it still requires each party to send 65 group elements (and the run-time is proportionately high).

In this paper we show that more efficient publicly-verifiable CCA2-encryption schemes can be obtained by using a novel concept of *relatively-sound* NIZKs rather than using simulation-sound NIZKs. Simulation-Sound NIZKs were first defined in [26], where it was used to convert Naor-Yung [24] CCA1-encryption scheme into a CCA2 encryption scheme. In simulation-sound NIZKs the NIZK (public) verifier continues to be sound even when the Adversary is given the simulated CRS and proofs. We notice that in most applications what is really required is that a (private) verifier with access to a trapdoor continues to be sound in the simulated world, as long as this private verifier is equivalent to the public verifier in the real-world[2].

We next show that an augmented El-Gamal encryption scheme (reminiscent of [11]), along with a labeled single theorem relatively-sound NIZK leads to a publicly-verifiable CCA2-encryption scheme. In the augmented El-Gamal scheme the public key (under the DDH or SXDH assumptions) consists of g, g^a, g^k, and the encryption of m with randomness x is $g^x, g^{ax}, m \cdot g^{kx}$. The labeled relatively-sound NIZK proves that the first two elements of the ciphertext use the same randomness x, with the third element used as label.

While a single theorem simulation-sound NIZK could also have been used above, we show that one can obtain single theorem relatively-sound NIZK far more cheaply than simulation-sound NIZK for this language. We use the fact that the language is a finite diverse group, and hence allows simple 2-universal projective hash functions [10], which allows us to build a private verifier. Under the SXDH assumption [16], converting a NIZK for this language to a relatively-sound NIZK only requires two more group elements, whereas the best-known simulation-sound extension would require nine group elements. Similarly, under the DLIN assumption, our extension requires only three more elements, whereas a simulation-sound extension requires at least 18 more elements [19]. Overall under the DLIN assumption, our publicly-verifiable CCA2 ciphertexts have only 19 group elements versus the 47 group elements in the Sahai scheme [26].

We show that using the new encryption scheme in the PAKE protocol of [19], leads to a new protocol which is two to three times more efficient (under both SXDH and DLIN assumptions), with the SXDH-based scheme requiring only 10 group elements to be communicated.

UC Security. Canetti et al [8] proposed a definition of security for password-based key exchange protocols within the Universally Composable (UC) security framework [7], which has the benefit of the universal composition theorem and as such can be deployed as a part of larger security contexts. In addition, their definition of security considers the case of arbitrary and unknown password distributions.

Katz and Vaikuntanathan [19] also gave a single round UC-secure protocol for password-based authenticated key exchange. However, their single round UC protocol is still inefficient as it uses general purpose NIZKs (for NP languages), and further requires proof of knowledge NIZKs. Even if the language for which zero knowledge proofs are required can be made to be given by simple algebraic relations in bilinear groups, the proof of knowledge for exponents of elements as required

[2]This verification trapdoor should not be same as the simulation trapdoor, as for this definition to be useful, the simulated proofs should be indistinguishable from the real proofs even when the Adversary has access to the verification trapdoor.

in their protocol makes it rather expensive[3].

A second main contribution of this paper is an efficient UC-secure single-round protocol for password based key exchange. The main new ideas required for this efficient protocol are as follows: (a) The shared secret key is obtained in the target group of the bilinear pairings used in the NIZKs which allows for efficient simulator-extraction of group elements corresponding to the smooth-hash trapdoor keys. Such an extraction is required for UC-simulatability. (b) The NIZK proof of knowledge (for extraction) requires the NIZKs to be unbounded simulation-sound. A general construction for unbounded simulation-soundness was given in [6] which is based on a construction due to Groth [15], both of which can be seen to be using relative-soundness implicitly. This leads us to give an optimized version of this general construction. (c) We continue to use the Damgard style [11] encryption scheme, which allows for even more optimization of the unbounded simulation-sound construction for this specific language.

As a result, we get a single-round UC-secure protocol, where under the DLIN-assumption, each party only communicates 63 group elements, which is as efficient as the PAK-model protocol described in [19]. Under the SXDH assumption, our UC-secure protocol only requires 33 group elements.

For sake of exposition, we focus on giving complete proofs only under the SXDH assumption. All of the protocols are also given under the DLIN assumption in the Appendix. The proofs extend naturally to DLIN, as has been demonstrated for the Cramer-Shoup CCA2-encryption scheme in [6].

2 NIZK Definitions

In this section we give some definitions related to Non Interactive Zero Knowledge (NIZK) proofs. We will assume familiarity with usual definitions of NIZKs (see e.g. [26, 16]). A proof for a relation R consists of a key generation algorithm K which produces the CRS ψ, a probabilistic polynomial time (PPT) prover P and a PPT verifier V.

Zero-Knowledge. We call (K, P, V) a **NIZK** proof for R if there exists a polynomial time simulator (S_1, S_2), such that for all non-uniform PPT adversaries $\mathcal{A}$ we have

$$\Pr[\psi \leftarrow K(1^m) : \mathcal{A}^{P(\psi,\cdot,\cdot)}(\psi) = 1] \approx \Pr[(\sigma,\tau) \leftarrow S_1(1^m) : \mathcal{A}^{S(\sigma,\tau,\cdot,\cdot)}(\sigma) = 1],$$

where $S(\sigma, \tau, x, w) = S_2(\sigma, \tau, x)$ for $(x, w) \in R$ and both oracles output failure if $(x, w) \notin R$.

One-time Simulation Soundness A NIZK proof is one-time simulation sound NIZK if for all non-uniform PPT adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ we have

$$\begin{aligned}\Pr[(\sigma,\tau) \leftarrow S_1(1^m); (x,s) \leftarrow \mathcal{A}_1(\sigma); \pi \leftarrow S_2(\sigma,\tau,x); (x',\pi') \leftarrow \mathcal{A}_2(x,\pi,\sigma,s) :\\ ((x',\pi') \neq (x,\pi)) \text{ and } \neg\exists w' \text{ s.t. } (x',w') \in R, \text{ and } V(\sigma,x',\pi') = 1] \approx 0.\end{aligned}$$

Labeled One-time Simulation Soundness. In a labeled NIZK, the prover takes an input label, in addition to the statement to be proven. Thus, the label acts as a context. The verifier takes a

[3]There are no known efficient NIZK proof of knowledge of exponents in the standard model, i.e. logarithms, including the Groth-Sahai system [23].

proof, a statement and a label. A labeled NIZK is one-time simulation sound if we have

$$\Pr[(\sigma,\tau) \leftarrow S_1(1^m); (x, label, s) \leftarrow \mathcal{A}_1(\sigma); \pi \leftarrow S_2(\sigma,\tau,x,label); (x', label', \pi') \leftarrow \mathcal{A}_2(x, label, \pi, \sigma, s) :$$
$$((x',\pi',\text{label}') \neq (x,\pi,\text{label})) \text{ and } \neg\exists w' \text{ s.t. } (x',w') \in R, \text{ and } V(\sigma, x', label', \pi') = 1] \approx 0.$$

Unbounded Simulation Sound Extractability (uSS-NIZK). Consider a NIZK proof (K, P, V, S_1, S_2) along with an initialization algorithm SE_1 and a knowledge extractor E_2, such that SE_1 outputs (σ,τ,ξ) with (σ,τ) identical to values output by S_1. Such a proof is said to have the Unbounded Simulation Sound Extractability property if for all non-uniform PPT adversaries $\mathcal{A}$ we have

$$\Pr[(\sigma,\tau,\xi) \leftarrow \text{SE}_1(1^k); (x,\pi) \leftarrow \mathcal{A}^{S_2(\sigma,\tau,\cdot)}(\sigma); w \leftarrow E_2(\sigma,\xi,x,\pi) \ :$$
$$(x,\pi) \notin Q \text{ and } (x,w) \notin R \text{ and } V(\sigma,x,\pi) = 1] \approx 0$$

where Q is the set of simulation queries and responses (x_i, π_i). For some subset of witnesses the extractor E_2 may extract witnesses in polynomial time, which will be the focus in this paper. For example, in Groth-Sahai NIZKs, bilinear group elements may be extractable in polynomial time.

2.1 Relative Soundness

We now define a novel *weaker notion of simulation soundness*, which might suffice for most applications, especially in the case of single theorem (or one-time) simulation. It is possible that this weaker notion may be more efficient to implement, as we demonstrate later for a particularly important language, where we also show that the weaker notion suffices for the application at hand. In a nutshell, the weaker notion allows for the simulator to have a private verifier of its own, with access to a trapdoor. Simulation soundness is now defined with respect to this simulator's private verifier, and hence the name *relative-soundness*. There is a further stipulation in the definition which states that in a hybrid world where the common reference string is same as in the real world but with the simulator having access to a private verification trapdoor, the public verifier is equivalent to the private verifier. In addition the zero-knowledge property should hold even when the Adversary is given the private verification trapdoor.

Single Theorem Relatively-Sound NIZK (1-SRS-NIZK). Consider a sound and complete proof (K, P, V) for a relation R along with a PPT private-verifier (W_1, W_2) and a PPT simulator (S_1, S_2), such that $W_1(1^m)$ outputs (ψ,ξ) with ψ identically distributed[4] to $K(1^m)$. Suppose V and W_2 are deterministic TMs. Such a proof is called a **single theorem relatively-sound NIZK** for R if for all non-uniform PPT adversaries $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4)$ we have

- $\Pr[(\psi,\xi) \leftarrow W_1(1^m); (x,\pi) \leftarrow \mathcal{A}_0(\psi) : V(\psi,x,\pi) \equiv W_2(\psi,\xi,x,\pi)] \approx 1$, **and**
- $\Pr[(\psi,\xi) \leftarrow W_1(1^m); (x,w,s) \leftarrow \mathcal{A}_1(\psi); \pi \leftarrow P(\psi,x,w) : \mathcal{A}_2(\xi,\pi,s) = 1] \approx$
  $\Pr[(\sigma,\tau,\xi) \leftarrow \text{S}_1(1^m); (x,w,s) \leftarrow \mathcal{A}_1(\sigma); \pi \leftarrow S_2(\sigma,\tau,x) : \mathcal{A}_2(\xi,\pi,s) = 1]$,
  where A_1 is restricted to producing (x,w) in R, **and**
- $\Pr[(\sigma,\tau,\xi) \leftarrow \text{S}_1(1^m); (x,s) \leftarrow \mathcal{A}_3(\sigma); \pi \leftarrow S_2(\sigma,\tau,x); (x',\pi') \leftarrow \mathcal{A}_4(x,\pi,\sigma,s) :$
  $((x',\pi') \neq (x,\pi)) \text{ and } \neg\exists w' \text{ s.t. } (x',w') \in R, \text{ and } W_2(\sigma,\xi,x',\pi') = 1] \approx 0.$

[4]This can be generalized to computational indistinguishability.

The variable s in the definitions designates a local state that the Adversary may maintain. The second condition says that the (one-time or single-theorem) zero-knowledge property[5] holds even when the Adversary is given the private-verifier trapdoor ξ. It is worth noting that the definition does not preclude the private-verifier trapdoor ξ and the simulation trapdoor τ from being dependent. Further, the verification trapdoor ξ produced by W_1 and S_1 need not be identically distributed, but just computationally indistinguishable to A_2.

The above definition is only accurate when the Verifiers V and W_2 are deterministic. When the verifiers are probabilistic, the first requirement should be

$$\Pr_{(\psi,\xi)\leftarrow W_1(1^m);(x,\pi)\leftarrow\mathcal{A}_0(\psi)}[|\Pr[V(\psi,x,\pi)] \approx W_2(\psi,\xi,x,\pi)|] \approx 1,$$

where the inner probabilities are over the local coins of the Verifiers. However, see the next subsection for an even weaker definition.

A **labeled** version of single theorem relative-soundness can also be defined just as above.

2.1.1 Alternate (weaker) definition of Relative Soundness

One can consider an even weaker definition of relatively-sound NIZKs which might suffice for most applications. In this definition, the Adversary A_2 is not given the verification trapdoor, but is given oracle access to the verifiers. Thus, zero-knowledge is now defined as indistinuishability of a real proof (cum real CRS) from a simulated proof (cum simulated CRS) even when the adversary has oracle access to the respective verifiers. This definition has the advantage that one does not need bullet one of the earlier definition. However, it is quite possible that for many languages the only way to go about proving this alternate relative-soundness property is to go via a hybrid W_1 as in the previous definition. On the other hand, it is also possible that there are efficient proofs for languages for which only this alternate definition can be proven (and a likely candidate is the dual-system concept of Waters [27]).

Alternate Definition. Consider a sound and complete proof (K, P, V) for a relation R along with a PPT private-verifier W_2 and a PPT simulator (S_1, S_2). Such a proof is called a **single theorem relatively-sound NIZK** for R if for all non-uniform PPT adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4)$ we have

- $\Pr[(\psi) \leftarrow K(1^m); (x, w, s) \leftarrow \mathcal{A}_1(\psi); \pi \leftarrow P(\psi, x, w) : \mathcal{A}_2^{V(\psi,\cdot,\cdot)}(\pi, s) = 1] \approx$
  $\Pr[(\sigma, \tau) \leftarrow \mathrm{S}_1(1^m); (x, w, s) \leftarrow \mathcal{A}_1(\sigma); \pi \leftarrow S_2(\sigma, \tau, x) : \mathcal{A}_2^{W_2(\sigma,\tau,\cdot,\cdot)}(\pi, s) = 1]$,
  where A_1 is restricted to producing (x, w) in R, **and**
- $\Pr[(\sigma, \tau) \leftarrow \mathrm{S}_1(1^m); (x, s) \leftarrow \mathcal{A}_3(\sigma); \pi \leftarrow S_2(\sigma, \tau, x); (x', \pi') \leftarrow \mathcal{A}_4(x, \pi, \sigma, s) :$
  $((x', \pi') \neq (x, \pi))$ and $\neg\exists w'$ s.t. $(x', w') \in R$, and $W_2(\sigma, \tau, x', \pi') = 1] \approx 0$.

Note that the verification trapdoor and the simulation trapdoor have been integrated into a single trapdoor τ.

[5] One may also consider generalizations where the zero-knowledge property is multi-theorem.

3 Smooth Projective Hash Functions

Fix a cyclic group $G = \langle g, \cdot \rangle$ of prime order q, such that $1/q$ is a negligible function of the security parameter. We define the El-Gamal encryption function. For A, K, m in G, and x, define

$$\mathsf{enc}_K^{\mathsf{eg}}(m; x) \;=\; \langle g^x, K^x \cdot m \rangle$$

For K and pwd in G, define $L_{K,\mathrm{pwd}} = \{c = \langle R, P \rangle \mid \exists x : c = \mathsf{enc}_K^{\mathsf{eg}}(\mathrm{pwd}; x)\} \cap G \times G$.

A **projective hash function** [10] is a keyed family of functions mapping elements in some message space X to the group G, and is associated with a language. Further, it comes with a **projection function** $\alpha : K \to S$, where K is the key space and S is the projected key space.

For our hash family, the key space is $\mathbb{Z}_q^* \times \mathbb{Z}_q^*$, and the projected key space is G. The message space X is the space of ciphertexts. For $n, \hat{n}$ in $\mathbb{Z}_q^*$, c in G^2, and K, pwd in G, define the hash family $\mathcal{H}^{K,\mathrm{pwd}}$ associated with $L_{K,\mathrm{pwd}}$ by

$$\mathcal{H}_{n,\hat{n}}^{\mathrm{pwd}}(c = \langle R, P \rangle) = (P/\mathrm{pwd})^{\hat{n}} \cdot R^n \qquad\qquad \alpha^{K,\mathrm{pwd}}(n, \hat{n}) = g^n \cdot (K)^{\hat{n}}.$$

It is straightforward to see that, if $c = \mathsf{enc}_K^{\mathsf{eg}}(\mathrm{pwd}; x)$ for some x, then $\mathcal{H}_{n,\hat{n}}^{\mathrm{pwd}}(c) = \alpha^{K,\mathrm{pwd}}(n, \hat{n})^x$.

For any K and pwd in G, $\mathcal{H}^{K,\mathrm{pwd}}$ is said to be **smooth** [10] w.r.t. $L = L_{K,\mathrm{pwd}}$, if for any c' in G^2, but *not* in L, the statistical distance between the distribution of the pair $(\mathcal{H}_{n,\hat{n}}^{K,\mathrm{pwd}}(c'), \alpha^{K,\mathrm{pwd}}(n.\hat{n}))$ and the pair (g^{d1}, g^{d2}) is negligible, where $n, \hat{n}, d1, d2$ are chosen randomly and independently from Z_q^*. It is a simple exercise to see that $\mathcal{H}^{K,\mathrm{pwd}}$ is smooth with respect to $L_{K,\mathrm{pwd}}$.

We also define a projective hash function family associated with language L to be **2-universal** [10] if for all $s \in S$, $x, x' \in X$, and $\pi, \pi' \in G$ with $x \notin L \cup \{x'\}$, it holds that

$$\Pr_k[H_k(x) = \pi \mid H_k(x') = \pi' \;\wedge\; \alpha(k) = s] \leq 1/q.$$

4 Bilinear Assumptions

Throughout the paper, we use (bilinear) groups G_1, G_2, G_T each of prime order p, which allow efficiently computable $\mathbb{Z}_q$-bilinear pairing maps $e : G_1 \times G_2 \to G_T$.

SXDH: [16] The symmetric external decisional Diffie-Hellman (SXDH) assumption states that the decisional Diffie-Hellman (DDH) problem is hard in both groups G_1 and G_2. Since an efficiently-computable isomorphism between the two groups, along with the bilinear pairing, renders the DDH problem easy, it must be assumed that there is no such isomorphism between the groups.

DLIN: [4] In groups such that G_1 is same as G_2, the decisional linear (DLIN) assumption states that given $(\alpha\mathcal{P}, \beta\mathcal{P}, r\alpha\mathcal{P}, s\beta\mathcal{P}, t\mathcal{P})$ for random $\alpha, \beta, r, s \in \mathbb{Z}_p$, and arbitrary generator $\mathcal{P}$ of G_1, it is hard to distinguish between $t = r + s$ and a random t.

It is an easy exercise to see that both these assumptions imply the standard computational Diffie-Hallman (CDH) assumption in both G_1 and G_2.

5 A Publicly-Verifiable CCA2-Encryption Scheme

In this section we describe a CCA2-Encryption scheme [25] that has the property that a potential ciphertext can be publicly verified to be a valid ciphertext of some message. Note that Sahai [26] had previously given a publicly-verifiable CCA2-encryption scheme employing the Naor-Yung CCA1-scheme [24], but our scheme is simpler and more efficient.

One might be tempted to take the Cramer-Shoup encryption scheme, and extend the ciphertext by including a NIZK proof that the 2-universal smooth projective-hash [10] was correctly computed. However, since the NIZK scheme by itself maybe malleable, this may render the scheme insecure in the CCA2-model. There are two potential fixes to this: (a) make the NIZK single theorem simulation-sound, or (b) include the NIZK commitments to the witness in the projective-hash. While it is not that difficult to see that (a) may lead to a correct publicly-verifiable CCA2-scheme (just as in [26]), the second idea (b) may seem far-fetched.

We now show that it suffices to make the NIZK proof a labeled single-theorem *relatively-sound* NIZK, and further one just needs to prove in this NIZK that the Diffie-Hellman tuple in the ciphertext is well-formed, i.e. it is of the form g^x, g^{ax}. We later show that there exists a very efficient way to extend a single-theorem Groth-Sahai NIZK of this statement to be a relatively-sound proof, such that the resulting publicly-verifiable CCA2-scheme is just the idea (b) mentioned above.

To formally define publicly-verifiable CCA2-encryption schemes, one just extends the standard IND-CCA2 definition of encryption [1] with a public verification function V which takes the public key and a potential ciphertext as arguments, and it returns true iff the decryption function when supplied with the same ciphertext does not return "invalid ciphertext".

For given g, A, let the relation $\mathcal{R} = \{((\rho, \hat{\rho}), x) \mid \rho = g^x, \hat{\rho} = A^x\}$. We now define a *labeled* publicly-verifiable public-key encryption scheme DHENC as follows:

Key Generation: Generate $g, A \xleftarrow{\$} \mathcal{G}$, and $k \xleftarrow{\$} \mathbb{Z}_q^*$. Let $K = g^k$. Let ψ be the CRS for a Labeled 1-SRS-NIZK. The public key is (g, A, K, ψ) and the private key is k.

Encrypt: Given plaintext $m \in \mathcal{G}$, and label $\mathtt{l}$. Choose $x \xleftarrow{\$} \mathbb{Z}_q^*$. Let the triple $\langle \rho, \hat{\rho}, \gamma \rangle$ be $\langle g^x, A^x, mK^x \rangle$. Let π be a 1-SRS-NIZK proof of $((\rho, \hat{\rho}), x) \in \mathcal{R}$ with label $\gamma, \mathtt{l}$. The ciphertext is $(\rho, \hat{\rho}, \gamma, \pi)$.

Decrypt: Given ciphertext $c = (\rho, \hat{\rho}, \gamma, \pi)$ and label $\mathtt{l}$. Verify if π is a 1-SRS-NIZK proof for $(\rho, \hat{\rho})$ and label $\gamma, \mathtt{l}$. If verification fails output $\perp$. Otherwise output $m = \frac{\gamma}{\rho^k}$.

Verify: Given ciphertext $c = (\rho, \hat{\rho}, \gamma, \pi)$ and label $\mathtt{l}$. Verify if π is a 1-SRS-NIZK proof for $(\rho, \hat{\rho})$ and label $\gamma, \mathtt{l}$. If verification fails output false else output true.

Theorem 1 *The scheme* DHENC *is publicly-verifiable (labeled) IND-CCA2 secure.*

The full proof of this theorem can be found in Appendix A, but the main idea is that the decryption can be done as either γ/ρ^k, or as $\gamma/(\rho^{k'}\hat{\rho}^{k''})$, where the Simulator chooses the public key K as $g^{k'}A^{k''}$. The encryption oracle hides the message by employing DDH as follows: (1) The NIZK CRS in the original experiment is the binding-CRS, but now the simulator retains a private-verification

trapdoor. (2) The decryption oracle in the original experiment does a public verification of proofs in each adversarially supplied ciphertext, but now the simulator replaces them with private verifications, as they are equivalent in this setting by definition. (3) The NIZK CRS is switched to be the hiding CRS, and proof switched to a simulator generated proof. This is an indistinguishable change by definition of 1-SRS-NIZK, even though the decryption oracle uses the verification trapdoor. Note, x is no more used in the simulated proof. (4) The decryption is done as $\gamma/(\rho^{k'}\hat{\rho}^{k''})$, which is equivalent because of single theorem relative-soundness. (5) DDH is employed, as only g^a instead of a is being used in simulation. (6) The decryption is done as γ/ρ^k, which is again equivalent by relative-soundness. (7) the message in the encryption can be switched by pairwise independence in k. (8) Next we do all the above steps (2)-(6) in reverse.

6 Single Theorem Relatively-Sound NIZK for the DDH Language

Let G_1 and G_2 be two groups with a bilinear pairing $e : G_1 \times G_2 \to G_T$ and $|G_1| = |G_2| = |G_T| = q$, a prime number. Also assume that DDH is hard for both G_1 and G_2. Recall that this is the `SXDH` assumption. Let $L_{g,A}$ be the language: $\{(\rho, \hat{\rho}) \in {G_1}^2 \mid \exists x.\ \rho = g^x \wedge\ \hat{\rho} = A^x\}$, with g, A in G_1.

Note that this language is actually a cyclic group with generator $\langle g, A\rangle$, and forms a diverse group system [10]. In [10], Cramer and Shoup show how to obtain 2-univeral projective hash functions for such languages, and we use these hash functions for private-verification.

We construct a single-theorem NIZK proof system, with a private verification function for $L_{g,A}$, which is relatively-sound, as follows:

CRS Generation: Generate $\mathcal{P} \xleftarrow{\$} G_2$ and $u, v, d_1, d_2, e_1, e_2 \xleftarrow{\$} \mathbb{Z}_p$. Compute $(P, Q, R, S, \mathbf{d}, \mathbf{e}) = (\mathcal{P}, \mathcal{P}^u, \mathcal{P}^v, \mathcal{P}^{uv+1}, g^{d_1}A^{d_2}, g^{e_1}A^{e_2})$. The CRS is $\psi = (P, Q, R, S, \mathbf{d}, \mathbf{e})$. The first four elements are as in the Groth-Sahai NIZK for SXDH, and the last two are the projection keys for a 2-universal projective-hash for the DDH language (just as [10]), to be used in the relatively-sound system. The private verification trapdoor key is $\xi = (d_1, d_2, e_1, e_2)$.

Prover: Given witness x and candidate (g^x, A^x), construct proof as follows. Generate $s \xleftarrow{\$} \mathbb{Z}_p$. Compute $t \leftarrow H(g^x, A^x, Q^xP^s, S^xR^s)$, where H is a collision resistant hash function. Then compute: $(\beta, c_1, c_2, \theta, \phi, \chi) \leftarrow \left((\mathbf{de}^t)^x, Q^xP^s, S^xR^s, g^s, A^s, (\mathbf{de}^t)^s\right)$. Output proof $\pi = (\beta, c_1, c_2, \theta, \phi, \chi)$. The first element is a 2-universal projective-hash computed on the candidate with witness x. The last five elements can be interpreted as generated by the Groth-Sahai NIWI proof (which also happens to be a NIZK proof) for the language $\{\rho, \hat{\rho}, h \mid \exists x\ :\ \rho = g^x, \hat{\rho} = A^x, h = (\mathbf{de}^t)^x\}$, where t is a hash of ρ, $\hat{\rho}$, *and the commitment* to x in the NIWI system, i.e. Q^xP^s, S^xR^s.

Public Verify: Given $\pi = (\beta, c_1, c_2, \theta, \phi, \chi)$ as a candidate proof of $(\rho, \hat{\rho})$, compute $t \leftarrow H(\rho, \hat{\rho}, c_1, c_2)$. Then check the following equations:

$$\begin{pmatrix} e(g, c_1) \stackrel{?}{=} e(\rho, Q) \cdot e(\theta, P), & e(g, c_2) \stackrel{?}{=} e(\rho, S) \cdot e(\theta, R) \\ e(A, c_1) \stackrel{?}{=} e(\hat{\rho}, Q) \cdot e(\phi, P), & e(A, c_2) \stackrel{?}{=} e(\hat{\rho}, S) \cdot e(\phi, R) \\ e(\mathbf{de}^t, c_1) \stackrel{?}{=} e(\beta, Q) \cdot e(\chi, P), & e(\mathbf{de}^t, c_2) \stackrel{?}{=} e(\beta, S) \cdot e(\chi, R) \end{pmatrix}$$

Private Verify: Given $\pi = (\beta, c_1, c_2, \theta, \phi, \chi)$ as a candidate proof of $(\rho, \hat{\rho})$, compute $t \leftarrow H(\rho, \hat{\rho}, c_1, c_2)$. Then first do public verification and if that succeeds then check the following equation: $\beta \stackrel{?}{=} \rho^{d_1} \hat{\rho}^{d_2} (\rho^{e_1} \hat{\rho}^{e_2})^t$.

Theorem 2 *The above system is a single-theorem relatively-sound NIZK proof system for $L_{g,A}$.*

The theorem is proven in detail in Appendix B, but it is worth pointing out here that we use the fact that in Groth-Sahai NIZKs, once the commitments to the witnesses are fixed, there is a unique proof satisfying the linear equations of the type used in the above NIZK proof. This holds for both the SXDH and the DLIN assumptions.

It is easy to extend this proof system to a labeled version in the following way. Compute t as before, but additionally include the label. That is, compute t as $H(\rho, \hat{\rho}, c_1, c_2, label)$.

The 1-SRS-NIZK proof for DDH language above consists of six group elements. The 1-SRS-NIZK proof for the DLIN language (and under the DLIN assumption) given in Appendix F.1 consists of 15 group elements.

7 Secure Protocol in the PAK Model

In this section we present a password-based key exchange protocol secure in the PAK model of security due to Bellare, Pointcheval and Rogaway [2]. The model is described in detail in Appendix C.1.

7.1 A Single Round Protocol in the PAK Model

We instantiate the single-round scheme due to Katz and Vaikuntanathan [19] which is described in Figure 1, with a more efficient publicly-verifiable CCA-secure encryption scheme, which enables a more efficient hash proof as well. The CRS, the projective-hash private key (k_i) generation, the projected key (α) generation, the encryption function $\text{enc}(\cdot, \cdot)$, the public verification function, the public projective-hash (H) computation, and the private-hash computation used in the protocol in Figure 1 are described below.

We instantiate the encryption to be the labeled publicly-verifiable CCA2-Encryption that we propose in this paper. Concretely, based on the SXDH assumption, we let the CRS, encryption and public verification functions be the same as the scheme DHENC.

CRS: Generate public key $(g, A, K, P, Q, R, S, \mathbf{d}, \mathbf{e})$ for the publicly-verifiable CCA2-Encryption scheme DHENC. Set $pk \leftarrow (g, A, K, P, Q, R, S, \mathbf{d}, \mathbf{e})$.

Encryption: Given plaintext m and label $label$, encrypt using the labeled encryption function of DHENC. That is, generate $x, s \stackrel{\$}{\leftarrow} \mathbb{Z}_q^*$. Then compute $t \leftarrow H(g^x, A^x, Q^x P^s, S^x R^s, K^x \cdot m, label)$. The ciphertext then is $\left\langle g^x, A^x, K^x \cdot m, (\mathbf{d}^t)^x, Q^x P^s, S^x R^s, g^s, A^s, (\mathbf{d}^t)^s \right\rangle$

Public Verification: Given ciphertext $c = \langle \rho, \hat{\rho}, \gamma, \beta, c_1, c_2, \theta, \phi, \chi \rangle$ and label $label$, first compute $t \leftarrow H(\rho, \hat{\rho}, c_1, c_2, \gamma, label)$. Then check the six equations exactly as in the public verify of DHENC scheme.

PAK Protocol for Password-based Key Exchange.

$\text{CRS} = pk$

Party P_i		$\mathcal{A}$		Party P_j
$k_i \xleftarrow{\$} \text{Hash-}K;\ s_i \leftarrow \alpha(k_i)$				$k_j \xleftarrow{\$} \text{Hash-}K;\ s_j \leftarrow \alpha(k_j)$
$label_i \leftarrow (P_i, P_j, s_i)$	$\xrightarrow{label_i, C_i}$		$\xleftarrow{label_j, C_j}$	$label_j \leftarrow (P_j, P_i, s_j)$
$C_i \leftarrow \mathsf{enc}_{pk}(label_i, pw)$				$C_j \leftarrow \mathsf{enc}_{pk}(label_j, pw)$
	$\xleftarrow{label'_j, C'_j}$		$\xrightarrow{label'_i, C'_i}$	
Reject if C'_j is not a valid ciphertext with label $label'_j$.				Reject if C'_i is not a valid ciphertext with label $label'_i$.
$sk_i \leftarrow H_{k_i}(label'_j, C'_j, pw)$ $\cdot H_{k_j}(label_i, C_i, pw)$				$sk_j \leftarrow H_{k_j}(label'_i, C'_i, pw)$ $\cdot H_{k_i}(label_j, C_j, pw)$

Figure 1: Single-round PAK-Model Secure Password-based Authenticated Key Exchange

The projective-hash family used in this scheme is $\mathcal{H}^{\text{pw}}$ along with the projection function $\alpha^{K,\text{pw}}$ defined in Section 3, where K is from the public-key in the CRS above. Note that the input *label* to the hash function is ignored in $\mathcal{H}^{\text{pw}}$. Also, α does not depend on pw.

Theorem 3 *Assume the existence of SXDH-hard groups G_1 and G_2. Then the protocol in Figure 1 is secure in the PAK model.*

The proof of this theorem is same as the proof in [19], as we have modularized the various constructs required in that proof. The main idea is that once the CCA2-encryption scheme is publicly verifiable, then the smooth hash needs to be just over the language $L_{K,\text{pw}}$, which are CPA encryptions of password.

8 Secure Protocol in the UC Model

8.1 UC Functionality for password-based key exchange

The essential elements of the Universal Composability framework are summarized in Appendix D.1. We adopt the definition for password-based key exchange from Canetti et al [8]. The following description is a summary from [8]. The formal description is given in Figure 3 in Appendix D.1.

Like the key exchange functionality, if both participating parties are not corrupted, then they receive the same uniformly distributed session key and the adversary learns nothing of the key except that it was generated. However, if one of the parties is corrupted, then the adversary determines the session key. This power to the adversary is also given in case it succeeds in guessing the parties' shared password. Participants also detect when the adversary makes an unsuccessful attempt. If the adversary makes a wrong password guess in a given session, then the session is marked interrupted and the parties are provided random and independent session keys. If the

adversary makes a successful guess, then the session is marked compromised. If a session is marked fresh, this means that it is neither interrupted nor compromised. Such sessions between uncorrupted parties conclude with both parties receiving the same, uniformly distributed session key.

Joint state and multi-session extension. In practical scenarios, different sessions might use the same common reference string. Canetti and Rabin [9] introduced the formalism of "universal composability with joint state" to address this problem. In this formalism, a multi-session extension $\hat{\mathcal{F}}$ of the given functionality $\mathcal{F}$ has to be defined as follows: $\hat{F}$ runs multiple independent copies of $\mathcal{F}$, where the copies are distinguished via sub-session IDs (SSIDs). $\hat{F}$ expects every incoming message to have an ssid in addition to the regular sid. The JUC theorem [9] asserts that composing a protocol π that uses several independent instances of $\mathcal{F}$, with a single copy of a protocol ρ that realizes $\hat{F}$, preserves the security of π.

UC Protocol for Password-based Key Exchange in SXDH Groups

$$\text{CRS} = g, \mathcal{P}, A, K, \psi : \quad g, A, K \xleftarrow{\$} G_1 \quad \mathcal{P} \xleftarrow{\$} G_2 \quad \psi = \text{uSS-NIZK } \mathsf{CRS}$$

Party P_i		Adversary $\mathcal{A}$
Input $(\texttt{NewSession}, sid, \mathsf{ssid}, P_i, P_j, \text{pwd}, initiator/responder)$		
Choose $x_1, n_1, \hat{n}_1 \xleftarrow{\$} \mathbb{Z}_q^*$.		
Set $\rho_1 = g^{x_1}$, $\hat{\rho}_1 = (A)^{x_1}$, $\gamma_1 = \text{pwd} \cdot K^{x_1}$, $\eta_1 = g^{n_1}(K)^{\hat{n}_1}$,	$\xrightarrow{c_1,\eta_1,\pi_1}$	$\mathcal{A}$
Let $c_1 = \langle \rho_1, \hat{\rho}_1, \gamma_1 \rangle$, and		
$\pi_1 = \text{uSS-NIZK}_\psi(\rho_1, \hat{\rho}_1, \eta_1; x_1, \mathcal{P}^{n_1}, \mathcal{P}^{\hat{n}_1})$ with label $\langle P_i, P_j, \mathsf{ssid} \rangle$.		
	$\xleftarrow{c_2',\eta_2',\pi_2'}$	$\mathcal{A}$
Let $c_2' = \langle \rho_2', \hat{\rho}_2', \gamma_2' \rangle$.		
If any of $\rho_2', \hat{\rho}_2', \gamma_2', \eta_2'$ is not in $G_1 \backslash \{1\}$, or		
not uSS-NIZK-Verify$(\pi_2; \rho_2', \hat{\rho}_2', \eta_2')$ with label $\langle P_j, P_i, \mathsf{ssid} \rangle$		
set $\text{sk}_1 \xleftarrow{\$} G_T$,		
else compute $h_2' = (\frac{\gamma_2'}{\text{pwd}})^{\hat{n}_1}(\rho_2')^{n_1}$, $h_1 = (\eta_2')^{x_1}$, set $\text{sk}_1 = e(h_2' \cdot h_1, \mathcal{P})$.		
Output $(sid, \mathsf{ssid}, \text{sk}_1)$.		

Figure 2: Single round UC-secure Password-based Authenticated Key Exchange under SXDH Assumption. It assumes two different groups G_1 and G_2 of prime order q, with a $\mathbb{Z}_q$-bilinear map e from $G_1 \times G_2$ to G_T, another group of order q. Let g and $\mathcal{P}$ be generators of G_1 and G_2 resp. The shared password pwd is assumed to be in G_1, and ssid is assumed to be in $\mathbb{Z}_q^*$. The uSS-NIZK$(\rho, \hat{\rho}, \eta; x, N, \hat{N})$ is a labeled unbounded-simulation G_2-extractable NIZK proof of membership in the language $L = \{\rho, \hat{\rho}, \eta \mid \exists x, N, \hat{N} : \rho = g^x, \hat{\rho} = A^x, e(\eta, \mathcal{P}) = e(g, N)e(K, \hat{N})\}$

8.2 A Single Round UC-Secure Password-Based Key Exchange Protocol

For sake of exposition, we describe the protocol based on the SXDH (symmetric external Diffie-Hellman) assumption. In the appendix, we give the protocol under the Decisional Linear assumption (DLIN). The SXDH based protocol is given in Fig 2, and more details can be found in Appendix D.2.

The protocol uses labeled unbounded simulation sound G_2-extractable NIZKs. A more optimized version of such a general NIZK [10] is given in the Appendix in Section E. In fact, for the language for which a NIZK is required in the protocol, i.e. $L = \{\rho, \hat{\rho}, \eta \mid \exists x, N, \hat{N} : \rho = g^x, \hat{\rho} = A^x, e(\eta, \mathcal{P}) = e(g, N)e(K, \hat{N})\}$, we give a further optimization in Appendix E.1. Based on this optimized construction, the uSS-NIZK requires 29 group elements. A similar construction under the DLIN assumption, and for the DLIN based UC-secure PWKE-construction (see Appendix F.4) requires 54 group elements.

Theorem 4 *Assume the existence of a SXDH-hard group, a labeled unbounded simulation-sound G_2-extractable NIZK proof system. Then the protocol in Figure 2 securely realizes the $\widehat{\mathcal{F}}_{\text{PWKE}}$ functionality in the $\mathcal{F}_{\mathsf{crs}}$ hybrid model, in the presence of static corruption adversaries.*

In the next section we demonstrate a simulator which uses $\widehat{\mathcal{F}}_{\text{PWKE}}$ to simulate the protocol to an adversary, thus proving Theorem 4.

8.3 The Simulator for the UC Protocol

The trapdoor keys a, k for the CRS are chosen differently by the simulator. Instead of choosing a, k randomly from $\mathbb{Z}_q^*$, the simulator chooses a, k', k'' from $\mathbb{Z}_q^*$ and sets $k = k' + a \cdot k''$. It outputs $A = g^a$ and $K = g^k = g^{k'}(g^a)^{k''}$ as before. Note that this does not change the distribution of A and K, as $\mathbb{Z}_q^*$ is a field. (We will continue to write k for $k' + ak''$, except when the simulation in some experiments needs to be done with g^a, instead of a).

Simulator S also invokes the initialization phase SE_1 of the labeled uSS-NIZK (with security parameter m) to obtain (σ, τ, ξ). S then gives A, K, and σ to the real world adversary $\mathcal{A}$ as the *common reference string*. Thereafter, the simulator S interacts with the environment Z, the functionality $\widehat{\mathcal{F}}_{\text{PWKE}}$, and uses $\mathcal{A}$ as a subroutine. The messages between Z and $\mathcal{A}$ are just forwarded by S.

The main difference in the simulation of the real world parties is that S uses a dummy message μ instead of the real password which it does not have access to. Further, it generates all proofs using the NIZK simulator S_2 instead of real prover.

8.3.1 New Session: Sending a message to $\mathcal{A}$

On message (NewSession, *sid*, ssid, i, j, role) from $\widehat{\mathcal{F}}_{\text{PWKE}}$, S starts simulating a new session of the protocol Π for party P_i, peer P_j, session identifier ssid, and CRS $= (A, K, \psi)$. We will denote this session by (P_i, ssid). To simulate this session, S chooses x_1 at random, and sets c_1 ($= \langle \rho_1, \hat{\rho}_1, \gamma_1 \rangle$) to $\langle g^{x_1}, A^{x_1}, \mu \cdot K^{x_1} \rangle$. It also chooses hash keys $n_1, \hat{n_1}$ at random, and computes the smooth-hash projected key η_1 as in the real protocol as well. S obtains a fake NIZK proof π_1 using the simulator S_2 of the NIZK, and the CRS σ, and simulation trapdoor τ. It then hands c_1, η_1, π_1 to $\mathcal{A}$ on behalf of this session.

More succinctly, the simulator behavior is as follows:

$$x_1, n_1, \hat{n}_1, \xleftarrow{\$} \mathbb{Z}_q^* \tag{1}$$

$$c_1\ (= \langle \rho_1, \hat{\rho}_1, \gamma_1 \rangle)\ = \langle g^{x_1}, A^{x_1}, \mu \cdot K^{x_1} \rangle \tag{2}$$

$$\eta_1 = \alpha^{K,\mu}(n_1, \hat{n}_1) = g^{n_1 + k\hat{n}_1} \tag{3}$$

$$\pi_1 = \text{uSS-NIZK-SIM}_{\psi}(\rho_1, \hat{\rho}_1, \eta_1) \text{ with label } \langle P_i, P_j, \mathsf{ssid} \rangle. \tag{4}$$

8.3.2 On Receiving a Message from $\mathcal{A}$

On receiving a message c_2', η_2', π_2' from $\mathcal{A}$ intended for this session (P_i, ssid), the simulator S makes the real world protocol checks including verifying the NIZK proof using the NIZK-verifier. If any of the checks fail, it issues a TestPwd call to $\widehat{\mathcal{F}}_{\text{PWKE}}$ with the dummy password μ, followed by a NewKey call with a random session key, which leads to the functionality issuing a random and independent session key to the party P_i (regardless of whether the session was interrupted or compromised).

Otherwise, it computes pwd$'$ by decrypting c_2', i.e. setting it to $\gamma_2'/(\rho_2')^k$. If the message received from $\mathcal{A}$ is same as message sent by S on behalf of peer P_j in session ssid, then S just issues a NewKey call for P_i. Otherwise, S calls $\widehat{\mathcal{F}}_{\text{PWKE}}$ with (TestPwd, ssid, P_i, pwd$'$). Regardless of the reply from $\mathcal{F}$, it then issues a NewKey call for P_i with key computed as follows (*this is different from the real-world protocol.*). This has the effect that if the pwd$'$ was same as the actual pwd in $\widehat{\mathcal{F}}_{\text{PWKE}}$ then the session key is determined by the Simulator, otherwise the session key is set to a random and independent value. Here is the complete simulator code (stated as it's overall experiment with Z, including $\mathcal{F}$'s communication with Z):

1. Let $c_2' = \langle \rho_2', \hat{\rho}_2', \gamma_2' \rangle$.
2. If any of $\rho_2', \hat{\rho}_2', \gamma_2', \eta_2'$ is not in $G_1 \backslash \{1\}$, or *not* uSS-NIZK-Verify$(\pi_2'; \rho_2', \hat{\rho}_2', \eta_2')$ with label $\langle P_j, P_i, \mathsf{ssid} \rangle$, output $\text{sk}_1 \xleftarrow{\$} G_T$, else compute as follows.
3. If msg rcvd == msg sent in same session (same SSID) by peer, set $\text{sk}_1 \xleftarrow{\$} G_T$, unless the peer also received a legitimate message and its key has already been set, in which case that same key is used to set sk_1.
4. Else, compute $N_2', \hat{N}_2'$ from the proof π_2', using the G_2-extraction trapdoor ξ.
5. Compute $\text{pwd}' = \gamma_2'/(\rho_2')^k$. If (pwd$'$ $\neq$ pwd) then $\text{sk}_1 \xleftarrow{\$} G_T$, else
6. $h_2' = (\frac{\gamma_2'}{\text{pwd}'})^{\hat{n}_1}(\rho_2')^{n_1},\ h_1 = (\eta_2')^{x_1}$; set $\text{sk}_1 = e(h_2', \mathcal{P}) \cdot e(h_1, \mathcal{P}) \cdot e(\mu/\text{pwd}, \hat{N}_2')$.

Note that the main difference is the additional factor $e(\mu/\text{pwd}, \hat{N}_2')$.

In Appendix D.3, we describe a series of experiments between the Simulator and the environment, starting with Expt_0 which is the same as the experiment described as the Simulator in this section, and ending with an experiment which is identical to the real world execution of the protocol. We will show that the environment has negligible advantage in distinguishing between these experiments, leading to a proof of Theorem 4.

References

[1] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In Hugo Krawczyk, editor, *Advances in Cryptology – CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 26–45. Springer, August 1998. 5, A

[2] Mihir Bellare, David Pointcheval, and Phillip Rogaway. Authenticated key exchange secure against dictionary attacks. In Bart Preneel, editor, *Advances in Cryptology – EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 139–155. Springer, May 2000. 1, 7, C.1

[3] Steven M. Bellovin and Michael Merritt. Encrypted key exchange: Password-based protocols secure against dictionary attacks. In *1992 IEEE Symposium on Security and Privacy*, pages 72–84. IEEE Computer Society Press, May 1992. 1

[4] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matthew Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55. Springer, August 2004. 1, 4

[5] Victor Boyko, Philip D. MacKenzie, and Sarvar Patel. Provably secure password-authenticated key exchange using Diffie-Hellman. In Bart Preneel, editor, *Advances in Cryptology – EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 156–171. Springer, May 2000. 1

[6] Jan Camenisch, Nishanth Chandran, and Victor Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In Antoine Joux, editor, *Advances in Cryptology – EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 351–368. Springer, April 2009. 1, E, 1, 2

[7] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd Annual Symposium on Foundations of Computer Science*, pages 136–145. IEEE Computer Society Press, October 2001. 1, D.1

[8] Ran Canetti, Shai Halevi, Jonathan Katz, Yehuda Lindell, and Philip D. MacKenzie. Universally composable password-based key exchange. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 404–421. Springer, May 2005. 1, 8.1

[9] Ran Canetti and Tal Rabin. Universal composition with joint state. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 265–281. Springer, August 2003. 8.1

[10] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars R. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 45–64. Springer, April / May 2002. 1, 3, 5, 6, 8.2, F, F.1

[11] Ivan Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In Joan Feigenbaum, editor, *Advances in Cryptology – CRYPTO'91*, volume 576 of *Lecture Notes in Computer Science*, pages 445–456. Springer, August 1992. 1

[12] Rosario Gennaro and Yehuda Lindell. A framework for password-based authenticated key exchange. In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 524–543. Springer, May 2003. `http://eprint.iacr.org/2003/032.ps.gz`. 1

[13] Oded Goldreich and Yehuda Lindell. Session-key generation using human passwords only. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 408–432. Springer, August 2001. `http://eprint.iacr.org/2000/057`. 1

[14] Li Gong, T. Mark A. Lomas, Roger M. Needham, and Jerome H. Saltzer. Protecting poorly chosen secrets from guessing attacks. *IEEE Journal on Selected Areas in Communications*, 11(5):648–656, June 1993. 1

[15] Jens Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In Xuejia Lai and Kefei Chen, editors, *Advances in Cryptology – ASIACRYPT 2006*, volume 4284 of *Lecture Notes in Computer Science*, pages 444–459. Springer, December 2006. 1

[16] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 415–432. Springer, April 2008. 1, 2, 4, B, 1, E

[17] Shai Halevi and Hugo Krawczyk. Public-key cryptography and password protocols. *ACM Transactions on Information and System Security*, 2(3):230–268, August 1999. 1

[18] Jonathan Katz, Rafail Ostrovsky, and Moti Yung. Efficient password-authenticated key exchange using human-memorable passwords. In Birgit Pfitzmann, editor, *Advances in Cryptology – EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 475–494. Springer, May 2001. 1, C.1

[19] Jonathan Katz and Vinod Vaikuntanathan. Round-optimal password-based authenticated key exchange. In Yuval Ishai, editor, *TCC 2011: 8th Theory of Cryptography Conference*, volume 6597 of *Lecture Notes in Computer Science*, pages 293–310. Springer, March 2011. 1, 7.1, 7.1, 6, F.3

[20] Chae Hoon Lim and Pil Joong Lee. Another method for attaining security against adaptively chosen ciphertext attacks. In Douglas R. Stinson, editor, *Advances in Cryptology – CRYPTO'93*, volume 773 of *Lecture Notes in Computer Science*, pages 420–434. Springer, August 1994. A

[21] Yujun Liu, Yonggang Cui, and Limin Liu. A publicly verifiable encryption scheme with short public/private keys. In *WASA'10*, pages 261–265, 2010. 1

[22] Philip D. MacKenzie, Sarvar Patel, and Ram Swaminathan. Password-authenticated key exchange based on RSA. In Tatsuaki Okamoto, editor, *Advances in Cryptology – ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 599–613. Springer, December 2000. 1

[23] S. Meiklejohn. An Extension of the Groth-Sahai Proof System. Brown University Masters Thesis, 2009. 3

[24] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd Annual ACM Symposium on Theory of Computing*. ACM Press, May 1990. 1, 5

[25] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Joan Feigenbaum, editor, *Advances in Cryptology – CRYPTO'91*, volume 576 of *Lecture Notes in Computer Science*, pages 433–444. Springer, August 1992. 5

[26] Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th Annual Symposium on Foundations of Computer Science*, pages 543–553. IEEE Computer Society Press, October 1999. 1, 2, 5

[27] Brent Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In Shai Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 619–636. Springer, August 2009. 2.1.1

A Appendix: Publicly-Verifiable CCA2 Encryption

Theorem 1 The scheme DHENC is publicly-verifiable IND-CCA2 secure.

The definition of IND-CCA2 security is now standard and can be found in [1]. In the labeled version [20], the Adversary calls the encryption oracle with two messages m_0, and m_1 and a label `l`. The encryption oracle returns c, an encryption of m_b with label l, with b chosen randomly. The Adversary can make arbitrary decryption calls, except that after receiving the encryption c, it cannot call the decryption oracle with c and the same label l. It is however allowed to call the decryption oracle with the same ciphertext c, but a different label l'. The notion of security is the advantage of the adversary in guessing b.

Proof: The public-verification part is straightforward, and we focus on the IND-CCA2 security part. We demonstrate a sequence of experiments showing indistinguishability of an encryption of an arbitrary m with an encryption of a fixed dummy message μ also in the message space. All experiments are identical to the previous experiment except for the noted modifications.

$\mathbf{Expt}_0$: This is the same as the real protocol, where if a ciphertext and label pair, i.e. $(\rho, \hat{\rho}, \gamma, \pi, \mathtt{l})$ is same as encryption query output and encryption query input label pair, then decryption oracle does not decrypt. Otherwise, decryption of a message $(\rho, \hat{\rho}, \gamma, \pi)$ with label `l`, after verifying the 1-SRS-NIZK proof π with label $(\gamma, \mathtt{l})$, is $m = \frac{\gamma}{\rho^k}$.

$\mathbf{Expt}_1$: In this experiment, the simulator generates the 1-SRS-NIZK CRS using W_1 and retains the verification-trapdoor ξ.

$\mathbf{Expt}_1$ is indistinguishable from $\mathbf{Expt}_0$ as W_1 generates the same CRS as real-world CRS generator K.

$\mathbf{Expt}_2$: In this experiment, the simulator chooses $k', k'' \xleftarrow{\$} \mathbb{Z}_q^*$ and sets $K = g^{k'} A^{k''}$ in the public key. It decrypts a message $(\rho, \hat{\rho}, \gamma, \pi)$, after verifying π with label $(\gamma, \mathtt{l})$, as $m = \frac{\gamma}{\rho^{k'} \hat{\rho}^{k''}}$.

The soundness of the 1-SRS-NIZK implies that $\hat{\rho} = \rho^a$ whp and hence $\rho^k = \rho^{k'} \hat{\rho}^{k''}$. Also the distribution of k remains the same. $\mathbf{Expt}_2$ is therefore indistinguishable from $\mathbf{Expt}_1$.

$\mathbf{Expt}_3$: This is same as $\mathbf{Expt}_2$, except that the simulator now uses the private verification function for proofs produced by the adversary. By the 1-SRS-NIZK property, private verification is equivalent to the public verification function. $\mathbf{Expt}_3$ is therefore indistinguishable from $\mathbf{Expt}_2$.

$\mathbf{Expt}_4$: In this experiment, the simulator switches to a CRS generated by S_1 and simulated proof of membership of (g^x, A^x) generated by S_2. That is, given challenge plaintext m, it returns the ciphertext $c = \Big(g^x, A^x, g^{k'x} A^{k''x} \cdot m, \text{Simulated } \pi\Big)$. Because of the switching, x is not needed as a witness in the 1-SRS-NIZK proof. Decryption query handling remains the same.

$\mathbf{Expt}_3$ is indistinguishable from $\mathbf{Expt}_4$, since the Adversary and the decryption simulation using the private verifier W_2 are polynomial time. Further, 1-SRS-NIZK requires that simulation is indistinguishable even if the Adversary has the verification trapdoor (which W_2 does indeed have, and in this game the decryption procedure is treated as part of the Adversary).

$\mathbf{Expt}_5$: In this experiment, the simulator replaces A^x by the DDH challenge $A^{x'}$. Therefore the response to the encryption challenge m becomes

$c = \Big(g^x, A^{x'}, g^{k'x} A^{k''x'} \cdot m, \text{Simulated } \pi\Big)$.

$\mathbf{Expt}_5$ is computationally indistinguishable from $\mathbf{Expt}_4$ due to the DDH assumption.

$\mathbf{Expt}_6$: In this experiment, the simulator decrypts as follows: on a message $(\rho, \hat{\rho}, \gamma, \pi)$ and label $\mathtt{l}$, after private-verifying the 1-SRS-NIZK π with label $(\gamma, \mathtt{l})$, output $m = \frac{\gamma}{\rho^k}$.

Note that the Experiment continues to ignore decryption queries if $(\rho, \hat{\rho}, \gamma, \pi, \mathtt{l})$ is same as encryption query output and encryption query input label pair. Thus, the 1-SRS-NIZK relative-soundness property implies that $\hat{\rho} = \rho^a$ whp and hence $\rho^k = \rho^{k'} \hat{\rho}^{k''}$. Hence, $\mathbf{Expt}_6$ is indistinguishable from $\mathbf{Expt}_5$.

$\mathbf{Expt}_7$: In this experiment, the simulator transforms k', k'' to $\tilde{k}', \tilde{k}''$, such that K remains same, that is, $k' + ak'' = \tilde{k}' + a\tilde{k}''$ $(= k)$, but at the same time $g^{k'x + ak''x'} \cdot m = g^{\tilde{k}'x + a\tilde{k}''x'} \cdot \mu$. This is always possible if $x \neq x'$, which is true with probability $1/q$.

Therefore the response to the encryption challenge m becomes

$c = \Big(g^x, A^{x'}, g^{\tilde{k}'x} A^{\tilde{k}''x'} \cdot \mu, \text{Simulated } \pi\Big)$.

Decryption of a message $(\rho, \hat{\rho}, \gamma, \pi)$, after verifying the 1-SRS-NIZK, is $m = \frac{\gamma}{\rho^k}$.

$\mathbf{Expt}_7$ is indistinguishable from $\mathbf{Expt}_6$ information theoretically.

$\mathbf{Expt}_8$: In this experiment, the simulator decrypts as follows: On a message $(\rho, \hat{\rho}, \gamma, \pi)$ and label $\mathtt{l}$, after private-verifying the 1-SRS-NIZK with label $\gamma, \mathtt{l}$, output $m = \frac{\gamma}{\rho^{\tilde{k}'} \hat{\rho}^{\tilde{k}''}}$.

Note that the Experiment continues to ignore decryption queries if $(\rho, \hat{\rho}, \gamma, \pi, \mathtt{l})$ is same as encryption query output and encryption query input label pair. Say, the encryption query output on input label $\mathtt{l}^*$ is $(\rho^*, \hat{\rho}^*, \gamma^*, \pi^*)$. Let the i-th decryption query be $(\rho_i, \hat{\rho}_i, \gamma_i, \pi_i, \mathtt{l}_i)$. If $(\rho_i, \hat{\rho}_i, \pi_i, \langle \gamma_i, \mathtt{l}_1 \rangle)$ is same as $(\rho^*, \hat{\rho}^*, \pi^*, \langle \gamma^*, \mathtt{l}^* \rangle)$ then the deccryption query ignores this decryption request. Since $\langle \gamma, \mathtt{l} \rangle$ was the label used in the NIZK proof generation, it follows from the 1-SRS-NIZK relative-soundness property that $\hat{\rho} = \rho^a$ whp and hence $\rho^k = \rho^{\tilde{k}'} \hat{\rho}^{\tilde{k}''}$. Also the distribution of k remains the same. $\mathbf{Expt}_8$ is therefore indistinguishable from $\mathbf{Expt}_7$.

$\mathbf{Expt}_9$: In this experiment, the simulator replaces $A^{x'}$ back by A^x. Therefore the response to the encryption challenge m becomes

$c = \Big(g^x, A^x, g^{\tilde{k}'x} A^{\tilde{k}''x} \cdot \mu, \text{Simulated } \pi\Big)$.

$\mathbf{Expt}_9$ is indistinguishable from $\mathbf{Expt}_8$ due to the DDH assumption.

$\mathbf{Expt}_{10}$: In this experiment, the simulator switches back to the CRS generated by W_1, and proofs generated by real prover with witness x. Therefore the response to the encryption challenge m becomes

$c = \Big(g^x, A^x, g^{\tilde{k}'x} A^{\tilde{k}''x} \cdot \mu, \pi\Big)$.

$\mathbf{Expt}_{10}$ is indistinguishable form $\mathbf{Expt}_9$ by the zero-knowledge property of 1-SRS-NIZK, even though the private-verifier (and hence the Adversary in this game) has access to verification trapdoor.

$\mathbf{Expt}_{11}$: This is same as $\mathbf{Expt}_{10}$, except that the simulator now uses the public verification function for proofs produced by the adversary. By the relative soundness property of the NIZK, this is equivalent to the result of the private verification function. $\mathbf{Expt}_{10}$ is therefore indistinguishable from $\mathbf{Expt}_{11}$.

$\mathbf{Expt}_{12}$: In this experiment, the simulator just uses k instead of $\tilde{k}', \tilde{k}''$. Therefore the response to the encryption challenge m becomes $c = (g^x, A^x, K^x \cdot \mu, \pi)$. Decryption of a message $(\rho, \hat{\rho}, \gamma, \pi)$, after verifying the 1-SRS-NIZK, is $m = \frac{\gamma}{\rho^k}$.

The soundness of the 1-SRS-NIZK implies that $\hat{\rho} = \rho^a$ whp and hence $\rho^k = \rho^{\tilde{k}'} \hat{\rho}^{\tilde{k}''}$. Also the distribution of k remains the same. $\mathbf{Expt}_{12}$ is therefore indistinguishable from $\mathbf{Expt}_{11}$.

$\mathbf{Expt}_{13}$: In this experiment, the simulator generates the NIZK CRS using K instead of W_1.

Since the two CRS are same, $\mathbf{Expt}_{13}$ is indistinguishable from $\mathbf{Expt}_{12}$.

This proves that DHENC is CCA2 secure because the protocol is indistinguishable from Experiment 13, in which everything is same except that in the response to an encryption request a constant message is encrypted, instead of the request message.

□

B Appendix: Proof of Relatively-Sound NIZK

Theorem 2: The system in Section 6 is a single-theorem relatively-sound NIZK proof system for $L_{g,A}$.

Proof:

Completeness and Soundness. Since the first four tests in the Public Verification above are same as the tests in the Groth-Sahai NIWI for the language $L_{g,A}$, soundness follows from soundness of the NIWI system [16]. Completeness follows by simple verification.

Zero Knowledge. We construct a simulator as follows. Generate $\mathcal{P} \xleftarrow{\$} G_2$; and $u, v, d_1, d_2, e_1, e_2 \xleftarrow{\$} \mathbb{Z}_p$. Set CRS $\sigma = (P, Q, R, S, \mathbf{d}, \mathbf{e}) = (\mathcal{P}, \mathcal{P}^u, \mathcal{P}^v, \boxed{\mathcal{P}^{uv}, g^{d_1} A^{d_2}, g^{e_1} A^{e_2}})$. The simulation trapdoor τ is $\boxed{u, d_1, d_2, e_1, e_2}$. Note that this trapdoor and the private-verifier trapdoor ξ share the smooth-hash keys.

Given a candidate $(\rho, \hat{\rho})$, generate the proof as follows. Generate $s \xleftarrow{\$} \mathbb{Z}_p$ and compute $t \leftarrow H(\rho, \hat{\rho}, P^s, R^s)$. Then compute

$$\pi = (\beta, c_1, c_2, \theta, \phi, \chi) = \left(\rho^{d_1} \hat{\rho}^{d_2} (\rho^{e_1} \hat{\rho}^{e_2})^t, P^s, R^s, \rho^{-u} g^s, \hat{\rho}^{-u} A^s, \beta^{-u} (\mathbf{d}\mathbf{e}^t)^s\right)$$

We now show that this CRS and a proof for $(\rho, \hat{\rho}) = (g^x, A^x) \in L_{g,A}$ is computationally indistinguishable from a real CRS and a real proof for the same candidate, given the DDH assumption

for G_2, even when the Adversary has access to the private verification trapdoor ξ. To this end, we enumerate a sequence of games each indistinguishable from the previous starting from a real CRS and a real proof and ending at the simulated CRS and a simulated proof:

Game 1: This is the real CRS and real proof:

$$\text{CRS } \psi = (P, P^u, P^v, P^{uv+1})$$

Proof for $(x : \rho, \hat{\rho}) = (x : g^x, A^x)$ can be seen to be computed as follows:

Generate $s \leftarrow \mathbb{Z}_p$
Compute $c_1 \leftarrow P^{ux+s}, c_2 \leftarrow (P^{uv+1})^x (P^v)^s$
Compute $t \leftarrow H(\rho, \hat{\rho}, c_1, c_2)$
Compute $\beta \leftarrow (\mathbf{d}\mathbf{e}^t)^x, \theta \leftarrow g^s, \phi \leftarrow A^s, \chi \leftarrow (\mathbf{d}\mathbf{e}^t)^s$
Proof $\pi = (\beta, c_1, c_2, \theta, \phi, \chi)$

Game 2: In this game the CRS is modified as follows:

$$\text{CRS } \sigma = (P, P^u, P^v, \boxed{P^{uv}})$$

By DDH for G_2, this is computationally indistinguishable from Game 1 (note the verification trapdoor ξ is independent of the DDH tuple).

Proof for $(x : \rho, \hat{\rho}) = (x : g^x, A^x)$ is computed identically as Game 1, except it uses the CRS σ:

Generate $s \leftarrow \mathbb{Z}_p$
Compute $c_1 \leftarrow P^{ux+s}, c_2 \leftarrow (P^{uv})^x (P^v)^s$
Compute $t \leftarrow H(\rho, \hat{\rho}, c_1, c_2)$
Compute $\beta \leftarrow (\mathbf{d}\mathbf{e}^t)^x, \theta \leftarrow g^s, \phi \leftarrow A^s, \chi \leftarrow (\mathbf{d}\mathbf{e}^t)^s$
Proof $\pi = (\beta, c_1, c_2, \theta, \phi, \chi)$

Game 3: In this game the NIZK CRS remains σ. The simulator is given the trapdoor u, d_1, d_2, e_1, e_2.

The proof is computed as follows:

Generate $s' \leftarrow \mathbb{Z}_p$
Compute $c_1 \leftarrow P^{s'}, c_2 \leftarrow (P^v)^{s'}$
Compute $t \leftarrow H(\rho, \hat{\rho}, c_1, c_2)$
Compute $\beta \leftarrow \rho^{d_1} \hat{\rho}^{d_2} (\rho_1^{e_1} \hat{\rho}^{e_2})^t, \theta \leftarrow \rho^{-u} g^{s'}, \phi \leftarrow \hat{\rho}^{-u} A^{s'}, \chi \leftarrow \beta^{-u} (\mathbf{d}\mathbf{e}^t)^{s'}$
NIZK Proof $\pi = (\beta, c_1, c_2, \theta, \phi, \chi)$

The proof for $(x : \rho, \hat{\rho}) = (x : g^x, A^x)$ is identical to Game 2, except that it uses the variable $s' = ux + s$. Observe that s' is distributed uniformly given uniform distribution of s. Also $g^s = \rho^{-u} g^{s'}$ and $A^s = \hat{\rho}^{-u} A^{s'}$.

Indistinguishability from Game 2 is information theoretic. Since Game 3 is just the simulator game, we are through.

Single-theorem Relative-Soundness. Suppose the adversary is provided a simulated proof $\pi = (\beta, c_1, c_2, \theta, \phi, \chi)$ for a language candidate $(\rho, \hat{\rho})$ chosen by the adversary. We have to show that any candidate and proof $(\rho', \hat{\rho'}, \pi') \neq (\rho, \hat{\rho}, \pi)$ provided by the adversary such that $(\rho', \hat{\rho'}) \notin L_{g,A}$, will fail the private verification test. Let $\pi' = (\beta', c'_1, c'_2, \theta', \phi', \chi')$ and $t' = H(\rho', \hat{\rho}', c'_1, c'_2)$. We analyze two cases:

Case $t' \neq t$: If the private verification succeeds, then we have the following linear equations, where $A = g^a, \rho = g^{x_1}, \hat{\rho} = g^{x_2}, \rho' = g^{x'_1}, \hat{\rho}' = g^{x'_2}$:

$$\begin{pmatrix} 1 & a & 0 & 0 \\ 0 & 0 & 1 & a \\ x_1 & x_2 & tx_1 & tx_2 \\ x'_1 & x'_2 & t'x'_1 & t'x'_2 \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \\ e_1 \\ e_2 \end{pmatrix} = \begin{pmatrix} \log \mathbf{d} \\ \log \mathbf{e} \\ \log \beta \\ \log \beta' \end{pmatrix}$$

Observe that if $t' \neq t$ and $x'_2 \neq ax'_1$, then the left-hand-side four by four matrix is full-ranked. Since, d_1, d_2, e_1, e_2 are chosen randomly and independently, this implies that $\log \beta'$ is uniformly distributed even conditioned on $\log \mathbf{d}, \log \mathbf{e}$ and $\log \beta$. Since, the latter three constitute the only information available to the adversary about d_1, d_2, e_1, e_2, the adversary succeeds in providing the correct β' only with negligible probability.

Case $t' = t$: In this case, by the collision resistance property of the hash, whp $(\rho', \hat{\rho}', c'_1, c'_2) = (\rho, \hat{\rho}, c_1, c_2)$. If private verification succeeds (which includes public verification), then

$$\begin{aligned} &\beta' = \rho^{d_1} \hat{\rho}^{d_2} (\rho^{e_1} \hat{\rho}^{e_2})^t = \beta \\ &e(\theta', P) = e(g, c_1) \cdot e(\rho, Q)^{-1} = e(\theta, P) \\ &e(\phi', P) = e(A, c_1) \cdot e(\hat{\rho}, Q)^{-1} = e(\phi, P) \\ &e(\chi', P) = e(\mathbf{d}\mathbf{e}^t, c_1) \cdot e(\beta', Q)^{-1} = e(\mathbf{d}\mathbf{e}^t, c_1) \cdot e(\beta, Q)^{-1} = e(\chi, P) \end{aligned}$$

Since the group G_T is a prime order group, using the bi-linearity of e, we get $\pi' = \pi$. Since $(\rho', \hat{\rho}') = (\rho, \hat{\rho})$ as well, we get a contradiction.

□

Labeled Single Theorem Relatively-Sound NIZK System It is easy to extend this proof system to a labeled version in the following way. Compute t as before, but additionally include the label. That is, compute t as $H(\rho, \hat{\rho}, c_1, c_2, label)$. Completeness, Soundness and Zero Knowledge proofs remain the same. For relative-soundness we again consider the two cases:

$t' = t$: In this case the label is forced to be the same whp due to the collision resistance property of H.

$t' \neq t$: This analysis remains the same as before.

C Appendix: Key Exchange in the PAK Model

C.1 PAK Model of Security

We describe a definition of security, which we refer to as PAK model in this paper, due to Bellare, Pointcheval and Rogaway [2]. This description summarizes a version depicted in [18].

Participants, passwords, and initialization. Prior to any execution of the protocol there is an initialization phase during which public parameters are established. We assume a fixed set User of protocol participants. For every distinct $U, U' \in \mathsf{User}$, we assume U and U' share a password $pw_{U,U'}$. We make the simplifying assumption that each $pw_{U,U'}$ is chosen independently and uniformly randomly from the set $[D] \stackrel{\text{def}}{=} \{1, \ldots, D\}$ for some integer D.

Execution of the protocol. We denote instance i of user U as Π_U^i.

- $\mathsf{sid}_U^i, \mathsf{pid}_U^i$, and sk_U^i denote the *session key*, *partner id* and *session key* for an instance, respectively.
- acc_U^i and term_U^i are boolean variables denoting whether a given instance has accepted or terminated, respectively.

The adversary's interaction with the principals (more specifically, with the various instances) is modeled via access to *oracles* which we describe now:

- $\mathsf{Send}(U, i, \mathsf{msg})$ — This sends message msg to instance Π_U^i. This instance runs according to the protocol specification. The output of Π_U^i is given to the adversary.

  The adversary can "prompt" instance Π_U^i to initiate the protocol with partner U' by query $\mathsf{Send}(U, i, U')$. In response to this query, instance Π_U^i outputs the first message of the protocol.
- $\mathsf{Execute}(U, i, U', j)$ — If Π_U^i and $\Pi_{U'}^j$ have not yet been used, this oracle executes the protocol between these instances and gives the transcript of this execution to the adversary. This oracle call represents passive eavesdropping of a protocol execution.
- $\mathsf{Reveal}(U, i)$ — This outputs the session key sk_U^i, modeling leakage of session keys.
- $\mathsf{Test}(U, i)$ — A random bit b is chosen; if $b = 1$ the adversary is given sk_U^i, and if $b = 0$ the adversary is given a session key chosen uniformly from the appropriate space.

Partnering. Let $U, U' \in \mathsf{User}$. Instances Π_U^i and $\Pi_{U'}^j$ are *partnered* if: (1) $\mathsf{sid}_U^i = \mathsf{sid}_{U'}^j \neq \text{NULL}$; and (2) $\mathsf{pid}_U^i = U'$ and $\mathsf{pid}_{U'}^j = U$.

Correctness. If Π_U^i and $\Pi_{U'}^j$ are partnered then $\mathsf{acc}_U^i = \mathsf{acc}_{U'}^j = \text{TRUE}$ and $\mathsf{sk}_U^i = \mathsf{sk}_{U'}^j$.

Advantage of the adversary. An instance Π_U^i is *fresh* unless one of the following is true at the conclusion of the experiment:

1. at some point, the adversary queried $\mathsf{Reveal}(U, i)$;
2. or, at some point, the adversary queried $\mathsf{Reveal}(U', j)$, where $\Pi_{U'}^j$ and Π_U^i are partnered.

An adversary $\mathcal{A}$ *succeeds* if it makes a single query $\mathsf{Test}(U, i)$ to a fresh instance Π_U^i, and outputs a bit b' with $b' = b$. We denote this event by Succ. The *advantage* of the adversary $\mathcal{A}$ in attacking Π is given by $\text{ADV}_{\mathcal{A},\Pi}(k) \stackrel{\text{def}}{=} 2 \cdot \Pr[\mathsf{Succ}] - 1$, where the probability is taken over the random coins used during the course of the experiment (including the initialization phase).

Definition 5 *Protocol* Π *is a* secure protocol for password-based authenticated key exchange *if, for all dictionary sizes* D *and for all* PPT *adversaries* $\mathcal{A}$ *making at most* $Q(n)$ *on-line attacks, it holds that* $\text{ADV}_{\mathcal{A},\Pi}(n) \leq Q(n)/D + \mathsf{negl}(n)$.

C.2 Proof of Security of the PAK protocol

Theorem 6 ([19]) *The the protocol in Figure 1 is secure in the PAK model provided that the encryption scheme is CCA2 secure and the hash function is smooth projective with respect to the language of labeled encryption of the password.*

Proof. We construct a simulator for the protocol execution which information theoretically hides the password in messages sent to the adversary and the keys generated. Hence the only way for the adversary to influence the key generation non-trivially is to demonstrate knowlege of the password pwd.

The secret key sk for the CCA2-secure encryption scheme is given to the simulator. The main difference in the simulation from the protocol is that S uses a dummy message μ instead of the real password which it does not have access to.

C.2.1 Passive Execute Queries

When the adversary asks for the transcript of a session between parties P_i and P_j, the simulator uses encryptions of μ, instead of the password:

$$k_i \xleftarrow{\$} \text{Hash-}K;\ s_i \leftarrow \alpha(k_i) \tag{5}$$

$$label_i \leftarrow (P_i, P_j, s_i) \tag{6}$$

$$C_i \xleftarrow{\$} \mathsf{enc}_{pk}(label_i, \mu) \tag{7}$$

$$\texttt{Send}\ (label_i, C_i) \tag{8}$$

Similarly for the party P_j. The key of both parties for this session is set to a common random value. We show that this transcript is indistinguishable from the real protocol by outlining a sequence of games starting from the real transcript and ending with the simulated transcript. Suppose P_j generates the message $(label_j, C_j) = (P_j, P_i, s_j, C_j)$.

Game 0: This is just the real protocol. We note that P_i computes its key as follows: $sk_i \leftarrow privH_{k_i}(label_j, C_j, \text{pwd}) \cdot pubH_{s_j}(label_i, C_i, \text{pwd}; x_i)$. The private hash algorithm $privH$ uses the private hash key for P_i, whereas the public hash algorithm $pubH$ uses the public hash key received and the witness for the encryption produced by P_i, that is, x_i.

Game 1: This is the real protocol with the following change. Since s_j is the public hash key actually generated by a session of P_j, the simulator knows the corresponding private hash key k_j. In this game, P_i computes its key using private hash keys only. That is, $sk_i \leftarrow privH_{k_i}(label_j, C_j, \text{pwd}) \cdot privH_{k_j}(label_i, C_i, \text{pwd})$. The transcript remains identical to Game 0.

Game 2: This is Game 1 with the following change. C_i and C_j are both computed as encryptions of μ. This is indistinguishable from Game 1 due to CCA2 security of the encryption scheme and because the encryption randomness is not used anywhere.

Game 3: This is Game 2 with the following change. The key of both parties for this session is set to a common random value. In Game 2 the key computated by both peers was $privH_{k_i}(label_j, C_j, \text{pwd}) \cdot privH_{k_j}(label_i, C_i, \text{pwd})$. Since both the ciphertexts are encryptions of μ, the hashes are over non-members of the language associated with it. Also the private hash keys k_i and k_j are not used by any other sessions of any principal. Hence the generated key is distributed uniformly from random. Therefore Game 3 is indistinguishable from Game 2.

Since Game 3 is just the simulated transcript we are done with this part.

C.2.2 New Session: Sending a message to $\mathcal{A}$

Next we look at the active attack queries. S starts simulating a new session of the protocol Π for party P_i, peer P_j, and CRS $= pk$. To simulate this session, S uses encryptions of μ, instead of the password:

$$k_i \xleftarrow{\$} \text{Hash-}K;\ s_i \leftarrow \alpha(k_i) \tag{9}$$

$$label_i \leftarrow (P_i, P_j, s_i) \tag{10}$$

$$C_i \xleftarrow{\$} \mathsf{enc}_{pk}(label_i, \mu) \tag{11}$$

$$\texttt{Send}\ (label_i, C_i) \tag{12}$$

C.2.3 On Receiving a Message from $\mathcal{A}$

On receiving a message $(label'_j, C'_j) = (P_j, P_i, s'_j, C'_j)$ from $\mathcal{A}$ purportedly from P_j, intended for this session of P_i, the simulator S first public verifies C'_j to be a ciphertext with label (P_j, P_i, s'_j). If the checks fail, the session is aborted.

We call a tuple (P_j, P_i, s, C) *well formed* for this session of P_i, if C can be public verified to be a valid ciphertext with label *label*. We call a well formed tuple to be *legitimate* if there is an actual session of P_j which generated this message. Two sessions of principals P_i and P_j are called partnered if P_i and P_j are peers in both sessions, and if the session of P_i receives the message produced by the session of P_j and vice versa.

If (P_j, P_i, s'_j, C'_j) is legitimate for this session and this is a partnered session with peer P_j, then choose a key value uniformly and independently from random and set it as the key for both the sessions. If there is no partner for this session, then set its key to be uniformly and independently random.

If the message is not even well formed then, as in the real protocol, the session is terminated. However, if the message is well formed but not legitimate for this session, then the simulator behavior is as follows. Let p' be the decryption of C'_j with label (P_j, P_i, s'_j). If $p' = \text{pwd}$, then declare A to be successful and terminate. Otherwise set sk_i to be random.

Since the messages sent out are information theoretically independent of the password, the probability of the decryption equaling the password is at most the probability of guessing the password by the adversary. In the next section, we describe a sequence of games leading from the real protocol finally ending up at the simulator, which will complete the proof.

C.2.4 Proof of Indistinguishability for the simulator

Now we outline a sequence of games leading from the real protocol, which we denote $\mathbf{Expt}_0$ finally ending up at the simulator described above.

$\mathbf{Expt}_1$: In this experiment, just as in the real protocol, P_i computes the first message as follows: generate private hash key k_i and let $s_i \leftarrow \alpha(k_i)$ be the public hash key. Generate randomness x_i and compute an encryption of pwd with label $label_i = (P_i, P_j, s_i)$. That is, compute $C_i = \mathsf{enc}_{pk}(label_i, \text{pwd}; x_i)$. Send $(label_i, C_i)$ to the adversary A.

On receiving a message (P_j, P_i, s'_j, C'_j) from the adversary A in this session, just as in the real protocol, the key sk_i is set to the actual computed key in case the message is legitimate for this session. That is, $sk_i \leftarrow privH_{k_i}(label'_j, C'_j, \text{pwd}) \cdot pubH_{s'_j}(label_i, C_i, \text{pwd}; x_i)$. The private hash algorithm $privH$ uses the private hash key for P_i, whereas the public hash algorithm $pubH$ uses the public hash key received and the witness for the encryption produced by P_i, that is, x_i.

If the message is not even well formed then, as in the real protocol, the session is terminated. However, if the message is well formed but not legitimate for this session, then the behavior is different and as follows. Let p' be the decryption of C'_j with label (P_j, P_i, s'_j). If $p' = \text{pwd}$, then declare A to be successful and terminate. Otherwise set sk_i to be random.

In the case of $p' \neq \text{pwd}$, we have that $(label'_j, C'_j, \text{pwd})$ is not in the language of the smooth hash. Also note that the private hash key k_i is not used anywhere other than this session. Hence the private hash $privH_{k_i}(label'_j, C'_j, \text{pwd})$ is uniformly random, even conditioned on $s_i = \alpha(k_i)$. Hence this step is indistinguishable to A from the real key computation. Making A succeed if $p' = \text{pwd}$ can only increase the advantage of A.

$\mathbf{Expt}_2$: This experiment is the same as $\mathbf{Expt}_1$ with the following change. On receiving a message (P_j, P_i, s'_j, C'_j) from the adversary A in this session, the key sk_i is computed in a different way in case the message is legitimate for this session. Since the message is legitimate, the simulator knows the private hash key k'_j corresponding to s'_j. It now computes the second hash with the private key rather than the public key. That is, $sk_i \leftarrow privH_{k_i}(label'_j, C'_j, \text{pwd}) \cdot privH_{k'_j}(label_i, C_i, \text{pwd})$. Since the message was legitimate, the hash is over a language member and hence the computations are actually the same. Observe that now the encryption randomness x_i is not used anywhere.

$\mathbf{Expt}_3$: This experiment is the same as $\mathbf{Expt}_2$ with the following changes. P_i sends out a message where the password pwd has been substituted by μ: $C_i \leftarrow \mathsf{enc}_{pk}(label_i, \mu)$. The key generation is changed in this experiment as follows. For partnered sessions, a key is generated uniformly from random and is set as the key for both the sessions. For all other sessions which receive legitimate messages, the key is generated uniformly and independently from random. The experiment remains unchanged for sessions which do not receive legitimate messages.

The indistinguishability of Experiments 2 and 3 follow by a sequence of hybrid experiments, where we change, session by session, the message being sent out from using the password pwd to using μ. Consider the sessions in the order of sending the first message. In this order, the i-th session is changed as follows, resulting in $\mathbf{Expt}_{2,i}$. Suppose the session is of principal P and its peer is Q. We change the message being sent by P from $(label_i, C_i)$ to $(label_i, \hat{C}_i)$, where $\hat{C}_i = \mathsf{enc}_{pk}(label_i, \mu)$.

Consider the scenario where this session receives a legitimate message (P_j, P_i, s'_j, C'_j) - a message generated by the j-th session in the order. We have two cases depending on whether i is less than or greater than j. Consider the case where i is less than j. In this case, set the key of session i to be uniformly and independently random instead of being computed by the projective hashes. If session j receives $(P_i, P_j, s_i, \hat{C}_i)$, then set the key of session j to be the same as session i. The key of all other sessions which receive $(P_i, P_j, s_i, \hat{C}_i)$, and are greater than i in the order, are set to be uniformly and independently random. Consider the case where i is greater than j. In this case, the key of session i was already set when the session j was considered - we just retain that. The key of all other sessions which receive $(s_i, \hat{C}_i)$, and are greater than i in the order, are set to be uniformly and independently random.

We show that $\mathbf{Expt}_{2,i}$ and $\mathbf{Expt}_{2,i-1}$ are indistinguishable. The ciphertexts C_i and $\hat{C}_i$ are indistinguishable by the CCA2 security of the encryption scheme and the fact that the encryption randomness is not used in any other computation. If session j is the partnered session of session i and j is greater than i, then in $\mathbf{Expt}_{2,i}$ they both compute the key as $privH_{k_i}(label_j, C_j, \text{pwd}) \cdot privH_{k_j}(label_i, \hat{C}_i, \text{pwd})$. Observe firstly, $(label_i, \hat{C}_i, \text{pwd})$ is not in the language of the hash. Secondly, sessions other than i which use the private key k_j for computing the second hash in $\mathbf{Expt}_{2,i}$, are all greater in order than i. Hence they compute hash over language members only. Due to these two observations, sessions i and j compute the same key, which is uniformly and independently random with respect to all other sessions. If the partnered session j is less than i in the order, then the key computation remains unchanged.

Now consider a session $l \neq j$ greater than i in the order, which receives $(P_i, P_j, s_i, \hat{C}_i)$. In $\mathbf{Expt}_{2,i}$ it computes the key as $privH_{k_i}(label_l, C_l, \text{pwd}) \cdot privH_{k_l}(label_i, \hat{C}_i, \text{pwd})$. Again we have, $(label_i, \hat{C}_i, \text{pwd})$ is not in the language of the hash. Again, sessions which use the private key k_l for computing the second hash in $\mathbf{Expt}_{2,i}$, are all greater in order than i. Hence they compute hash over language members only. Due to these two observations, session l computes a key, which is uniformly and independently random with respect to all other sessions. If l is less than i in the order, then l's key computation remains unchanged.

We end up at $\mathbf{Expt}_3$ after all these hybrid experiments, and hence it is indistinguishable from $\mathbf{Expt}_2$. Since $\mathbf{Expt}_3$ is just the simulator we are through.

D Appendix: Key Exchange in the UC Model

D.1 Universally Composable Security

The Universally Composable (UC) framework [7] is a formal system for proving security of computational systems such as cryptographic protocols. The framework describes two probabilistic games: The *real world* that captures the protocol flows and the capabilities of an attacker, and the *ideal world* that captures what we think of as a secure system. The notion of security asserts that

these two worlds are essentially equivalent.

The real-world model. The players in the real-world model are all the entities of interest in the system (e.g., the nodes in a network, the processes in a software system, etc.), as well as *the adversary* A and *the environment* $\mathcal{Z}$. All these players are modeled as efficient, probabilistic, message-driven programs (formally, they are all interactive Turing machines).

The actions in this game should capture all the interfaces that the various participants can utilize in an actual deployment of this component in the real world. In particular, the capabilities of A should capture all the interfaces that a real-life attacker can utilize in an attack on the system. (For example, A can typically see and modify network traffic.) The environment $\mathcal{Z}$ is responsible for providing all the inputs to the players and getting all the outputs back from them. Also, $\mathcal{Z}$ is in general allowed to communicate with the adversary A. (This captures potential interactions where higher-level protocols may leak things to the adversary, etc.)

The ideal-world model. Security in the UC framework is specified via an "ideal functionality" (usually denoted $\mathcal{F}$), which is thought of as a piece of code to be run by a completely trusted entity in the ideal world. The specification of $\mathcal{F}$ codifies the security properties of the component at hand. Formally, the ideal-world model has the same environment as the real-world model, but we pretend that there is a completely trusted party (called "the functionality"), which is performing all the tasks that are required of the protocol. In the ideal world, participants just give their inputs to the functionality $\mathcal{F}$, which produces the correct outputs (based on the specification) and hands them back to the participants. $\mathcal{F}$ may interact with an adversary, but only to the extent that the intended security allows. (E.g., it can "leak" to the adversary things that should be publicly available, such as public keys.)

UC-Security. An implementation π **securely realizes** an ideal functionality $\mathcal{F}$ if no external environment can distinguish between running the protocol π in the real world and interacting with the trusted entity running the ideal functionality $\mathcal{F}$ in the ideal world. That is, for every adversary A in the real world, there should exist an adversary A' in the ideal world, such that no environment $\mathcal{Z}$ can distinguish between interacting with A and π in the real world and interacting with A' and $\mathcal{F}$ in the ideal world.

D.2 UC Password-Based KE-Protocol in SXDH

Given a security parameter m, let $G = \mathcal{G}^m$ be an asymmetric bilnear group of prime order q ($= poly(m)$). We will assume that there are three groups G_1^m, G_2^m and G_T^m, such that there is an efficiently computable bilinear pairing e from $G_1^m \times G_2^m$ to G_T^m. Further, we assume that the *Decisional Diffie-Hellman* is hard in both the group sequences $\{\mathcal{G}_1^m\}_{m\geq 1}$, and $\{\mathcal{G}_1^m\}_{m\geq 1}$. All three groups are assumed to be of the same prime order q. Our proposed protocol, formally described in Figure 2 runs as follows:

Consider parties P_i and P_j involved in the protocol with SSID ssid. The CRS is three group elements $g, A(= g^a), K(= g^k)$ chosen randomly from G_1, another element $\mathcal{P}$ chosen randomly from G_2, and a uSS-NIZK CRS ψ (a labeled unbounded-simulation sound G_2-extractable NIZK). Since g,

Functionality $\mathcal{F}_{\mathbf{pwKE}}$

The functionality $\mathcal{F}_{\text{pwKE}}$ is parameterized by a security parameter k. It interacts with an adversary S and a set of parties via the following queries:

Upon receiving a query $(\mathsf{NewSession}, sid, P_i, P_j, pw, role)$ **from party** P_i**:**
Send $(\mathsf{NewSession}, sid, P_i, P_j, role)$ to S. In addition, if this is the first NewSession query, or if this is the second NewSession query and there is a record (P_j, P_i, pw'), then record (P_i, P_j, pw) and mark this record fresh.

Upon receiving a query $(\mathsf{TestPwd}, sid, P_i, pw')$ **from the adversary** S**:**
If there is a record of the form (P_i, P_j, pw) which is fresh, then do: If $pw = pw'$, mark the record compromised and reply to S with "correct guess". If $pw \neq pw'$, mark the record interrupted and reply with "wrong guess".

Upon receiving a query $(\mathsf{NewKey}, sid, P_i, sk)$ **from** S**, where** $|sk| = k$**:**
If there is a record of the form (P_i, P_j, pw), and this is the first NewKey query for P_i, then:

- If this record is compromised, or either P_i or P_j is corrupted, then output (sid, sk) to player P_i.
- If this record is fresh, and there is a record (P_j, P_i, pw') with $pw' = pw$, and a key sk' was sent to P_j, and (P_j, P_i, pw) was fresh at the time, then output (sid, sk') to P_i.
- In any other case, pick a new random key sk' of length k and send (sid, sk') to P_i.

Either way, mark the record (P_i, P_j, pw) as completed.

Figure 3: The password-based key-exchange functionality $\mathcal{F}_{\text{pwKE}}$

$\mathcal{P}$ are also part of the uSS-NIZK CRS, having chosen the NIZK CRS, g, $\mathcal{P}$ are already determined. The protocol is symmetric and asynchronous with each party computing a message to be sent, then receiving a corresponding message and computing a key. Therefore, we just describe it from the perspective of one party; the other is symmetric.

Party P_i generates $x \xleftarrow{\$} Z_q^*$ and computes $c_1 = \langle g^x, A^x, K^x \cdot pw \rangle$. It also generates hash key $(n_1, \hat{n_1}) \xleftarrow{\$} (Z_q^*)^4$ and computes the projection key $\eta_1 = \alpha^{K,\text{pwd}}(n_1, \hat{n_1}) = g^n \cdot K^{\hat{n}}$. Finally it computes a NIZK proof of consistency in the following way:

$$\pi_1 = \text{uSS-NIZK}_\psi(g^x, A^x, \eta_1; x, \mathcal{P}^{n_1}, \mathcal{P}^{\hat{n}_1}) \text{ with label } \langle P_i, P_j, \mathsf{ssid} \rangle$$

Note that π here denotes the commitments to the witnesses as well as the further proof as in the Groth-Sahai system.

The NP language L for the NIZK is

$$L = \{\rho, \hat{\rho}, \eta \mid \exists x, N, \hat{N} \ : \ \rho = g^x, \hat{\rho} = A^x, \ e(\eta, \mathcal{P}) = e(g, N)e(K, \hat{N})\}$$

Now, the message sent by P_i is $\langle c_1, \eta_1, \pi_1 \rangle$. Let the message received by P_i in this session, supposedly from P_j, be $\langle c_2', \eta_2', \pi_2' \rangle$. Let c_2' be parsed as $(\rho_2', \hat{\rho_2'}, \gamma_2')$. If any of $\rho_2', \hat{\rho}_2', \gamma_2', \eta_2'$ is not in $G_1 \backslash \{1\}$, or uSS-NIZK-Verify$(\pi_2'; \rho_2', \hat{\rho}_2', \eta_2')$ with label $\langle P_j, P_i, \mathsf{ssid} \rangle$ turns out to be false, then it sets

its session key sk_1 randomly from the target group of e, G_T. Otherwise it is computed as follows:

$$h_2' = (\frac{\gamma_2'}{\mathrm{pwd}})^{\hat{n}_1}(\rho_2')^{n_1} \qquad h_1 = (\eta_2')^{x_1} \qquad h_3 = h_2' \cdot h_1 \qquad \mathrm{sk}_1 = e(h_3, \mathcal{P}).$$

D.3 Proof of Indistinguishability for the UC Protocol

We now describe a series of experiments between the Simulator and the environment, starting with Expt_0 which is the same as the experiment described as the Simulator in Section 8.3, and ending with an experiment which is identical to the real world execution of the protocol in Fig 2. We will show that the environment has negligible advantage in distinguishing between these experiments, leading to a proof of realization of $\mathcal{F}_{\mathrm{PWKE}}$ by the protocol Π described in Figure 2.

For each instance, we will use subscript 2 along with a prime, to refer to variables after the reception of the message from $\mathcal{A}$, and use subscript 1 to refer to variables computed before sending the message to $\mathcal{A}$. We will call a message legitimate if it was not altered by the adversary, and delivered in the correct session, and to the correct party.

Expt_1**:** The first major change in the simulation is to set sk_1 using smooth hash trapdoor keys in the case where the message was legitimate, but still using the wrong password μ instead of the real password pwd. Hence, because the hash proof system is for languages with messages encrypting real password, the smoothness yields random values from the adversary's point of view.

Thus, after these series of changes in the simulation the experiment Expt_1 is same as Expt_0 except for the following modified step 3 in the reception code: *If msg rcvd == msg sent in same session by peer, set* sk_1 to

$$e(\mathcal{H}^{\mathrm{pwd}}_{n_1,\hat{n}_1}(\mathsf{enc}^{\mathsf{eg}}_{g^k}(\mu; x_2)) \cdot \mathcal{H}^{\mathrm{pwd}}_{n_2,\hat{n}_2}(\mathsf{enc}^{\mathsf{eg}}_{g^k}(\mu; x_1)), \mathcal{P}).$$

Note that in showing that the above is same as a random value, we only employ the hash proof system corresponding to n_1 and $\hat{n}_1$, and note that the second factor corresponding to n_2 and $\hat{n}_2$ is independent of the first. In step 6, n_1 and $\hat{n}_1$ are being used, but the code never gets there if the msg received is same as message sent by legitimate peer.

Expt_2: Next, we replace all occurrences of $e(h_1, \mathcal{P})$ $(= e((\eta_2')^{x_1}, \mathcal{P}))$ in the computation of sk_1 in Step 6 of the reception code by $e(g, N_2')^{x_1} \cdot e(K, (\hat{N}_2')^{x_1})$, which is same as $e(g^{x_1}, N') \cdot e(K^{x_1}, \hat{N}')$. This leads to an indistinguishable change as the simulator had verified the NIZK proofs, and the NIZK proofs have unbounded simulation extractability property, and thus $e(\eta_2', \mathcal{P}) = e(g, N_2')e(K, \hat{N}_2')$.

Expt_3: The next change in simulation is to replace μ by the real password in the outgoing message element γ. However, since the simulator is employing k to compute pwd', one cannot directly employ DDH to replace μ by pwd in outgoing γ. However, since we are using an augmented El-Gamal encryption scheme, i.e. also including $\hat{\rho}$ in the outgoing message along with a proof of its relation to ρ, we can use the pairwise independence in k to accomplish our goal.

To be more precise, we will have a sequence of hybrid experiments, one for each session and party pair (ssid, P_i), where we replace μ by pwd in the outgoing γ of (ssid, P_i), as well as in all occurrences of μ in the reception phase of simulation of (ssid, P_i), i.e. on reception of a (supposed) message from the peer of (ssid, P_i). First in every session, we replace the step *compute* $\mathrm{pwd}' = \gamma_2'/\rho_2'^k$ by *compute* $\mathrm{pwd}' = \gamma_2'/(\rho_2'^{k'}\hat{\rho}_2'^{k''})$. This is indistinguishable because of simulation soundness of the

NIZK system, and the fact that the simulator has verified the NIZK proofs. We also replace all occurrences of K by $g^{k'}A^{k''}$, and all occurrences of K^{x_1} by $g^{x_1k'}A^{x_1k''}$ (this can be in the code of session (ssid, P_i) or its peer). Next, we employ the DDH property of the group G_1 to replace all occurrences of A^{x_1} by $g^{a'x_1}$ (i.e. employing DDH on g^{x_1} and g^a), where a' is another independent random variable. Next, we switch *compute* $\text{pwd}' = \gamma_2'/(\rho_2'^{k'}\hat{\rho}_2'^{k''})$ back to $\text{pwd}' = \gamma_2'/\rho_2'^{k}$, again by simulation soundness. At this stage, because of pairwise independence redundancy in k, we can replace all occurrences of $g^{k'x_1+k''a'x_1} \cdot \mu$ by $g^{k'x_1+k''a'x_1} \cdot \text{pwd}$, while keeping $K = g^{k'+ak''}$ intact. Next, we again replace *compute* $\text{pwd}' = \gamma_2'/\rho_2'^{k}$ by *compute* $\text{pwd}' = \gamma_2'/(\rho_2'^{k'}\hat{\rho}_2'^{k''})$. Employing DDH once again on g^{x_1} and g^a, we replace all occurrences of $g^{a'x_1}$ by g^{ax_1}. Finally, we switch back to $\text{pwd}' = \gamma_2'/\rho_2'^{k}$.

At this point, not only is the outgoing γ_1 being computed as $K^{x_1} \cdot \text{pwd}$, i.e. $c_1 = \mathsf{enc}^{\mathsf{eg}}_K(\text{pwd}; x_1)$, but also in the reception phase of the same (ssid, P_i), the term $e(\mu/\text{pwd}, \hat{N}_2')$ has been replaced by 1. Recall that in Expt_2, $e(h_1, \mathcal{P})$ was replaced by $e(g^{x_1}, N') \cdot e(K^{x_1}, \hat{N}')$, and now $e(K^{x_1}, \hat{N}')$ has been replaced by $e(\text{pwd}/\mu \cdot K^{x_1}, \hat{N}')$, which is then equivalent to replacing $e(\mu/\text{pwd}, \hat{N}_2')$ by 1 in Step 6.

Further, if the message received was legitimate, then sk_1 is now set to

$$e(\mathcal{H}^{\text{pwd}}_{n_1,\hat{n}_1}(\mathsf{enc}^{\mathsf{eg}}_{g^k}(\mu; x_2)) \cdot \mathcal{H}^{\text{pwd}}_{n_2,\hat{n}_2}(\mathsf{enc}^{\mathsf{eg}}_{g^k}(\text{pwd}; x_1), \mathcal{P}).$$

Similarly, if the peer received a legitimate message, its computation of sk_1 has a similar change, i.e. its first factor has μ replaced by pwd.

Thus, at the end of these sequence of hybrid experiments, if the message received was legitimate, then sk_1 is now set to

$$e(\mathcal{H}^{\text{pwd}}_{n_1,\hat{n}_1}(\mathsf{enc}^{\mathsf{eg}}_{g^k}(\text{pwd}; x_2)) \cdot \mathcal{H}^{\text{pwd}}_{n_2,\hat{n}_2}(\mathsf{enc}^{\mathsf{eg}}_{g^k}(\text{pwd}; x_1)), \mathcal{P}).$$

Expt_4: In this experiment we drop the condition *if (pwd$'$ $\neq$ pwd) then set sk_1 to random* in Step 5, and always output as follows

$$h_2' = (\frac{\gamma_2'}{\text{pwd}})^{\hat{n}_1/\mathsf{ssid}}(\rho_2')^{n_1},\ h_1 = (\eta_2')^{x_1};\ \text{set } \text{sk}_1 = e(h_2', \mathcal{P}) \cdot e(g^{x_1}, N_2') \cdot e(K^{x_1}, \hat{N}_2').$$

This is accomplished by a series of hybrid experiments, one for each (ssid, P_i), we employ the hash proof smoothness, as $\text{pwd}' \neq \text{pwd}$ implies the tuple c_2' is not in the language, and hence h_2' is anyway random and independent.

Expt_5: In this experiment we set sk_1 in the last step as

$$e(h_2', \mathcal{P}) \cdot e(\eta_2'^{x_1}, \mathcal{P}).$$

This change is indistinguishable as the simulator is checking the validity of the NIZK proofs, and by simulation-soundness extractability.

Expt_6: In this experiment we can drop the extraction of N_2' and $\hat{N}_2'$, as they are no longer needed.

Expt_7: In this experiment we do a syntactic change, by dropping step 3. Note that currently that step is computing sk_1 as

$$e(\mathcal{H}^{\text{pwd}}_{n_1,\hat{n}_1}(\mathsf{enc}^{\mathsf{eg}}_{g^k}(\text{pwd}; x_2)) \cdot \mathcal{H}^{\text{pwd}}_{n_2,\hat{n}_2}(\mathsf{enc}^{\mathsf{eg}}_{g^k}(\text{pwd}; x_1)), \mathcal{P}),$$

but since $\eta_2' = \eta_2$, and $c_2' = c_2$ for this session, then the above expression is same as

$$e(h_2', \mathcal{P}) \cdot e(\eta_2'^{x_1}, \mathcal{P}).$$

Expt_8: We replace all simulator generated proofs by proofs generated by real prover, and indistinguishability follows from the zero-knowledge property of the uSS-NIZK.

At the end of the experiment, we switch from the CRS generated by SE_1 to that generated by real-world NIZK initialization.

At this point, the experiment Expt_7 is indistinguishable from the real-world experiment, and the only property required to verify this claim is the completeness of the hash proof system, i.e. when the labelled tuple c, ssid is in the language, then the hash can be computed from the projection keys and the witness x_1 of c.

That completes the proof of Theorem 4. □

E More Efficient Unbounded Simulation Sound NIZKs

In [6], an unbounded simulation sound NIZK scheme is given for bilinear groups, building on the Groth-Sahai NIZKs and using Cramer-Shoup like CCA2 encryption schemes under K-linear assumptions. In this section we show various general optimizations for that construction, and further optimizations for specific languages involving generalized Diffie-Hellman tuples.

The general optimizations can be summarized as follows.

1. The scheme in [6] uses a one-time signature scheme. However, since it also uses a labeled CCA2 encryption scheme, the one-time signature scheme can be dropped, and one can use the label in the CCA2 scheme to get the signature property.

2. The scheme in [6] allows the simulator to generate a CCA2 encryption of u^x (for trapdoor x) along with a proof, instead of the proof of the statement. In order for the Adversary to cheat, it must also produce such an encryption, which is impossible under CCA2. However, one notices that since the simulator knows u^x, instead of a normal encryption, the simulator can hide u^x with just the smooth hash.

We now give this optimized version under the SXDH-assumption (for ease of exposition). Similar optimizations can be obtained under the DLIN assumption. Let the SXDH-group be (G_1, G_2, G_T), each of the groups of order q, with a $\mathbb{Z}_q$-bilinear map e from $G_1 \times G_2$ to G_T. We will write the bilinear map $e(A, B)$ in infix notation as $A \cdot B$. The group operation will be written in additive notation.

Languages for the simulation-sound NIZK can be specified by equations (relations) of the form $\vec{x} \cdot \vec{A} = T$, where $\vec{x}$ are variables from Z_q, $\vec{A}$ are constants from G_2, and T is a constant from G_T, or vice-versa, and thus $\vec{x}$ serves as witness for a member of a language specified by $\vec{A}$ and T. Languages can also be specified by equations of the form $\vec{B} \cdot \vec{\mathcal{Y}} = T_1 \cdot T_2$, where $\vec{B}$ are elements from G_1, $\vec{\mathcal{Y}}$ are variables from G_2, and T_1 and T_2 are constants from G_1 and G_2 resp. One can also consider languages with multiple such relations of both kinds.

Note that languages for which Groth-Sahai NIWI proofs can be given are more general, including equations like $\vec{x} \cdot \vec{A} + \vec{b} \cdot \vec{\mathcal{Y}} = T$, as well as quadratic equations.

The uss-NIZK CRS will consist of the usual Groth-Sahai NIWI CRS for SXDH, along with $g, A{=}g^a, \mathbf{k} = g^{k1}A^{k2}, \mathbf{d}{=}g^{d1}A^{d2}, \mathbf{e}{=}g^{e1}A^{e2}$, and $\mathbf{h}{=}g^x, \mathbf{u}{=}g^u$, with $g \in G_1$, and $a, k1, k2, d1, d2, e1, e2, x, u$ chosen at random from Z_q. One could alternatively choose these values from G_2. Let H be a collision resistant hash function.

Given a set of relations as above, along with satisfying variables, the prover does the following:

1.
   - For each equation of the kind $\vec{x} \cdot \vec{A} = T$, it generates a modified equation $\vec{x} \cdot \vec{A} = \delta \cdot T$, where δ is a new global integer variable.
   - Get modified equations of the form $\vec{B} \cdot \vec{\mathcal{Y}} + T_1 \cdot \mathcal{V} = 0$, where $\mathcal{V}$ is a new variable representing elements from G_2, along with an additional equation $\mathcal{V} + (\delta - 1) \cdot T_2 = 0$ [16].
   - Generate an additional quadratic equation $\delta(1 - \delta) = 0$.

2. Produce a Groth-Sahai NIWI proof for the above modified set of equations, with δ set to 1. Call this proof, which includes all commitments to original variables as well as δ and $\mathcal{V}$, as π_1. Also append the original statement to be proven in π_1.

3. Generate $\rho = g^w$, $\hat{\rho} = A^w$, with w chosen at random.

4. Produce a Groth-Sahai NIWI proof of the following statements (using the same commitment to δ as in step 2, and w', x' committed to zero): $\rho^{1-\delta} = g^{w'}$, $\hat{\rho}^{1-\delta} = A^{w'}$, $\mathbf{h}^{1-\delta} = g^{x'}$. Call this proof along with commitments to x', w' as π_2.

5. Set $b = \mathbf{u} \cdot (\mathbf{kde}^t)^w$, where $t = H(\rho, \hat{\rho}, \pi_1, \pi_2)$.

6. Produce a Groth-Sahai NIWI proof of the following statement (using the same commitment to δ as in step 2, and same commitment for w', x' as in Step 4): $b^{1-\delta} = \mathbf{u}^{x'} \cdot (\mathbf{kde}^t)^{w'}$. Call this proof π_3.

7. The uss-NIZK proof consists of $(\pi_1, \pi_2, \pi_3, \rho, \hat{\rho}, b)$.

Verification. The proof $(\pi_1, \pi_2, \pi_3, \rho, \hat{\rho}, b)$ of a statement S is verified by checking π_1 against the modified equations obtained from S, obtaining $t = H(\rho, \hat{\rho}, \pi_1, \pi_2)$, and verifying π_2 and π_3.

Completeness. By construction.

Soundness. Since δ is forced to be zero or one by soundness of the NIWI system, if an adversary can prove a false statement S', then δ must be zero. In such a case, the proof π_2, π_3 entails that there exists a w, x, such that $\rho = g^w, \hat{\rho} = A^w, b = \mathbf{u}^x \cdot (\mathbf{kde}^t)^w$, and $\mathbf{h} = g^x$. Since $(\mathbf{kde}^t)^w$ can be computed from ρ, $\hat{\rho}$, π_1, π_2, and the keys $k1$, $k2$, $d1$, $d2$, $e1$, $e2$, this allows the simulator to solve a CDH game with instance $(g, \mathbf{h}, \mathbf{u})$, by using this Adversary.

Zero Knowledge. Let the simulator be (S_1, S_2), where S_1 is the initialization routine. S_1 generates the CRS exactly as in the real-world but the Simulator S_2 gets the trapdoor x.

Before we describe how S_2 generates proofs, consider a hybrid prover, which works like the real prover, except computes $b = \mathbf{u}^x \cdot (\mathbf{kde}^t)^w$ in step 5, where $t = H(\rho, \hat{\rho}, \pi_1, \pi_2)$. By employing DDH on tuple $g^{k1}, \rho = g^w, g^{k1 \cdot w}$ in group G_1, it follows that this value is indistinguishable from one generated by real prover. Note that w is not used by the real prover as a witness in any proof.

The simulator S_2 sets $\delta = 0$, and sets all original variables to zero (note that since the groups $G1$ etc. are written in additive notation, zero is the unity of the group). Thus, it allows it to give

a NIWI proof π_1, indistinguishable from the hybrid prover. Next, it produces $\rho = g^w$, $\hat{\rho} = A^w$, with w chosen at random, thus same as in hybrid prover. By NIWI, the proof π_2 generated with $w' = w$, $x' = x$ and $\delta = 0$ is indistinguishable from π_2 generated by hybrid prover. Next, simulator generates $b = \mathbf{u}^x \cdot (\mathbf{kde}^t)^w$, where $t = H(\rho, \hat{\rho}, \pi_1, \pi_2)$, just as in the hybrid prover. This, allows the simulator to generate π_3 indistinguishable from the proof π_3 generated by hybrid prover (again, by NIWI).

Unbounded Simulation Soundness Extractability. First note that the simulator initialization algorithm S_1 generates the binding CRS for the NIWI system, which is same as the CRS for the real world. In such a case, it is a property of the Groth-Sahai NIWI that witnesses can be extracted from the commitments in the proof (although only group elements from G_1 and G_2 can be *efficiently* extracted).

Suppose the Adversary on obtaining proofs from the simulator S_2, manages to obtain a *new* proof π^* on a statement s^*, such that the extracted witnesses, along with s^* do not satisfy the relation R, whereas the verifier accepts. Thus, by soundness of the NIWI, it must be the case that $\delta = 0$ in this proof π^*. First note that if $\rho, \hat{\rho}, \pi_1, \pi_2$ are same in π^* as a S_2 generated proof, and further that b is also identical in the two proofs, then π_3 is guaranteed to be identical in the two proofs. This follows from a property of the Groth-Sahai NIWIs (both for SXDH and DLIN assumptions) that the proofs are uniquely determined by the commitments to the variables, for relations of the kind $\vec{x} \cdot \vec{A} = T$, and $b^{1-\delta} = \mathbf{u}^{x'} \cdot (\mathbf{kde}^t)^{w'}$ is indeed a relation of this kind.

Now, if $\rho, \hat{\rho}, \pi_1, \pi_2$ is same in the two proofs, and thus t is also same in the two proofs, but b is different, then since S_2 is encrypting $\mathbf{u}^x$ in b, with $\mathbf{h} = g^x$, the proof π_3 from π^* will be invalid. Thus, $\langle \rho, \hat{\rho}, \pi_1, \pi_2 \rangle$ in π^* must be different from every proof generated by S_2, and from the collision resistance property of H, we can assume that the t values are also different.

Let W be the event that a π^* is valid, but the relation R does not hold. We next show by a sequence of games that $\Pr[W]$ is negligible.

Game 1: The challenger generates the uss-NIZK CRS, and retains $a, k1, k2, d1, d2, e1, e2$, as well as x as trapdoors. The challenger generates proofs using S_2 on Adversary supplied statements. In other words it commits $\delta = 0$ and $x' = x$. The Adversary supplies a proof π^*. If $\langle \rho, \hat{\rho}, \pi_1, \pi_2 \rangle$ is same as some S_2 generated proof then output NO and halt. If t is same as the t value for some S_2 generated proof, output NO and halt. Otherwise, the challenger checks if

$$b/(\rho^{k1+d1+t\cdot e1} \hat{\rho}^{k2+d2+t\cdot e2}) = \mathbf{u}^x.$$

If so, it outputs YES, else it outputs NO. Let W_1 be the event that Challenger outputs YES. By soundness of the NIWI system and collision-resistance property of H, it follows that $\Pr[W]$ is at most $\Pr[W_1]$ plus a negligible amount - recall that under the event W, π^* has δ committed to 0, and $\langle \rho, \hat{\rho}, \pi_1, \pi_2 \rangle$ is distinct from values in all S_2 generated proofs.

Game 2: This is the same as Game 1, except that the challenger now uses the hiding CRS of the Groth-Sahai NIZK to generate proofs π_2 and π_3 with variables w' and x' committed to zero. Recall that Groth-Sahai NIWIs are also NIZKs for the equations of the type required in π_2 and π_3, where the ZK simulation uses an *implicit* δ', similar to the δ used in the above description. Let W_2 be the event that the challenger outputs YES in this game. By the ZK property of the Groth-Sahai NIZK, $|\Pr[W_1] - \Pr[W_2]|$ is negligible.

Game 3: By a sequence of hybrid games, one for each proof generated, the challenger changes b in each proof it generates to encrypt $\mathbf{u}^0$ instead of $\mathbf{u}^x$. Given that the challenger is decrypting b from π^*, one needs to employ the two-universal smooth hash property just as in the Cramer-Shoup CCA2-encryption scheme. In more details, in a first sub-hybrid game in the hybrid game corresponding to the generation of say the i-th proof, the challenger, indistinguishably under the DDH assumption, changes every occurrence of A^{w_i} to $g^{a \cdot w'_i}$ for a new and independent random variable w'_i. In the next hybrid game, it changes the test to $b/(\rho^{k1+a \cdot k2+d1+t \cdot e1} \hat{\rho}^{d2+t \cdot e2}) = \mathbf{u}^x$. This sub-hybrid game has almost the same probability of outputting YES as the previous sub-hybrid game, because if $\hat{\rho} \neq \rho^a$, then by the 2-universal smooth hash property, the above test would fail with high probability in both games, and hence both games would output NO in that case. In the next sub-hybrid game, one employs a pairwise-independent information theoretic argument involving $k1, k2$, to replace $\mathbf{u}^x$ by $\mathbf{u}^0$ in the i-th proof's b value. In the next sub-hybrid game, one switches back to the original test, and again as before by the 2-universal smooth hash property the two games are indistinguishable. Finally, one employs DDH again, to return to $g^{a \cdot w_i}$ for $\hat{\rho}$ in the i-th proof. Let W_2 be the event that the challenger outputs YES in this final hybrid game. By the arguments above, $|\Pr[W_2] - \Pr[W_1]|$ is negligible.

Now we split the challenger, into a CDH challenger generating $g, \mathbf{h}, \mathbf{u}$, and a CDH-solver generating the NIZK-CRS, conducting rest of the game 3 above using the CDH challenge, and incorporating the Adversary, and computing $b/(\rho^{k1+d1+t \cdot e1} \hat{\rho}^{k2+d2+t \cdot e2})$ on the Adversary's proof π^*, and returning this value to the CDH challenger. Note that $\Pr[W_2]$ is exactly the probability of the CDH-solver solving the CDH-challenge, which by assumption is negligible. Hence $\Pr[W]$ is negligible as well.

It is noteworthy that the uss-NIZK CRS can just give the product of $\mathbf{k}$ and $\mathbf{d}$, and it follows that $\mathbf{k}$ can be deleted altogether from the scheme.

The above can also be made a labeled unbounded simulation-sound extractable NIZK, by including the label in the collision-resistance hash computation t in step 5. The proof follows by a simple modification of the above proof.

Note that it takes 14 extra group elements to convert a NIZK proof into a uSS-proof using this construction. This follows from the description in [16] for the SXDH construction, where we need commitments to additional variables δ, ε', x', which require 2 group elements each, and proofs for a quadratic integer equation involving δ which requires 4 group elements, and four addional linear equations which require one group element each.

In the case of DLIN assumption, one would need 28 extra group elements.

For the language in Section 8.2, the NIZK proof requires commitments to x, $N, \hat{N}$ and $\mathcal{V}$. There are two linear equations, one pairing linear equation, and one multi-scalar multiplication equation involving δ and $\mathcal{V}$, which require a total of 4*2+2*1+2+6 = 18 group elements. In the next section, we show a further optimization for this language, which saves another 3 group elements, resulting in a total of 14+18-3 = 29 gtoup elements for the uss-NIZK proof for the language.

E.1 Further Optimization for Specific Languages

We now show that for languages which include Diffie-Hellman tuples as components, one can obtain further optimizations in the unbounded simulation sound NIZK proofs of Section E. One example of such a language is the one used in the UC-secure password-based key-exchange protocol of Fig 2. That language has two equations which require two components to be Diffie-Hellman tuples. If this tuple is with respect to g and A as in the uss-NIZK CRS, then we obtain further optimization as follows.

Thus, let the language have relations $\{f, \hat{f} \mid \exists y : f = g^y,\ \hat{f} = A^y\}$. Further assume that the witness y is not involved in any other relations defining the language.

Consider the proof being generated in steps 1-7 in Section E. Instead of generating proof π_1 on statements $g^y = f^\delta$, $A^y = \hat{f}^\delta$ as in step 2, skip this step for these two relations. All other relations are still proven as in Step 2 and included in π_1. Step 3 is as before. In Step 4, consider the following equations: $\rho^{1-\delta} f^\delta = g^{w'}$, $\hat{\rho}^{1-\delta} \hat{f}^\delta = A^{w'}$, $\mathbf{h}^{1-\delta} = g^{x'}$. Give a NIWI proof of this with δ as committed in Step 2 (where it is committed to 1), and w' committed to y. Include this commitment of w' in this proof π_2.

In step 5, set $b = \mathbf{u} \cdot (\mathbf{kde}^t)^w$, where $t = H(\rho, \hat{\rho}, \pi_1, \pi_2)$, and $\tilde{b} = (\mathbf{kde}^t)^y$.

In step 6, prove $b^{1-\delta} \tilde{b}^\delta = \mathbf{u}^{x'} \cdot (\mathbf{kde}^t)^{w'}$. Note, w' is committed to y. Call this proof π_3.

In step 7, output the uss-NIZK proof as $(\pi_1, \pi_2, \pi_3, \rho, \hat{\rho}, b, \tilde{b})$.

The proof that this constitutes an unbounded-simulation-sound NIZK for these languages is same as in Section E, but additionally noting that the Simulator can generate $\tilde{b}$ from f, $\hat{f}$, and its smooth-hash trapdoor keys.

This optimization saves on the commitment to y, and the two proofs for f and $\hat{f}$, while adding a new group element $\tilde{b}$. In SXDH, this leads to a saving of 3 group elements. A similar scheme in DLIN leads to a saving of nine group elements (3+3+2+2-1), recalling that in DLIN the generalized-Diffie-Hellman tuple is a triple with two witnesses, while the smooth-hash remains a single group element.

F Secure Protocols under DLIN Assumption

In this section, we instantiate the protocols under the DLIN assumption. Let G be a group with a bilinear pairing $e : G \times G \to G_T$ and $|G| = |G_T| = q$, a prime number. Also assume that DLIN is hard for G. Let $L_{g,f,h}$ be the language: $\{(\rho, \sigma, \tau) \in G^3 \mid \exists x, y.\ \rho = g^x \wedge\ \sigma = f^y \wedge\ \tau = h^{x+y}\}$, with g, f, h in G. The proofs are analogous to the SXDH versions and these generalizations can be obtained as in [10].

F.1 Single Theorem Relatively-Sound NIZK for the DLIN Language

We construct a single-theorem NIZK proof system, with a private verification function for $L_{g,f,h}$, which is relatively-sound, as follows:

CRS Generation: Generate $d_1, d_2, e_1, e_2, u_1, u_2 \xleftarrow{\$} \mathbb{Z}_p$ and $\tilde{\psi}$, a CRS for a Groth-Sahai NIWI under the DLIN assumption. Compute $(\mathbf{d}_1, \mathbf{e}_1, \mathbf{d}_2, \mathbf{e}_2) = (g^{d_1} h^{u_1}, f^{e_1} h^{u_1}, g^{d_2} h^{u_2}, f^{e_2} h^{u_2})$. The

CRS is $\psi = (\tilde{\psi}, \mathbf{d}_1, \mathbf{e}_1, \mathbf{d}_2, \mathbf{e}_2)$. The last four elements are the projection keys for a 2-universal projective-hash for the DLIN language (just as [10]), to be used in the relatively simulation sound system. The private verification trapdoor key is $\xi = (d_1, d_2, e_1, e_2, u_1, u_2)$.

Prover: Given witness x, y and candidate (g^x, f^y, h^{x+y}), construct proof as follows. Let *com* be Groth-Sahai commitments to exponents x, y. Compute $t \leftarrow H(g^x, f^y, h^{x+y}, com)$, where H is a collision resistant hash function. Then compute $\beta \leftarrow (\mathbf{d}_1 \mathbf{e}_1{}^t)^x (\mathbf{d}_2 \mathbf{e}_2{}^t)^y$. This is a 2-universal projective-hash computed on the candidate with witness x, y. Let $\tilde{\pi}$ be the Groth-Sahai NIWI proof (which also happens to be a NIZK proof) for the language $\{\rho, \sigma, \tau, \beta \mid \exists x, y : \rho = g^x, \sigma = f^y, \tau = h^{x+y}, \beta = (\mathbf{d}_1 \mathbf{e}_1{}^t)^x (\mathbf{d}_2 \mathbf{e}_2{}^t)^y$, where t is a hash of ρ, σ, τ, *and the commitment* to x, y in the NIWI itself. Output proof $\pi = (\beta, \tilde{\pi})$.

Public Verify: Given $\pi = (\beta, \tilde{\pi})$ as a candidate proof of (ρ, σ, τ), let *com* be the witness commitments part of $\tilde{\pi}$. Compute $t \leftarrow H(\rho, \sigma, \tau, com)$. Then check $\tilde{\pi}$ as a Groth-Sahai NIWI proof for the statement $\exists x, y \ : \rho = g^x, \sigma = f^y, \tau = h^{x+y}, \beta = (\mathbf{d}_1 \mathbf{e}_1{}^t)^x (\mathbf{d}_2 \mathbf{e}_2{}^t)^y$

Private Verify: Given $\pi = (\beta, \tilde{\pi})$ as a candidate proof of (ρ, σ, τ), let *com* be the witness commitments part of $\tilde{\pi}$. Compute $t \leftarrow H(\rho, \sigma, \tau, com)$. Then first do public verification and if that succeeds then check the following equation: $\beta \stackrel{?}{=} \left(\rho^{d_1} \sigma^{e_1} \tau^{u_1}\right) \left(\rho^{d_2} \sigma^{e_2} \tau^{u_2}\right)^t$.

Theorem 7 *The above system is a single-theorem relatively-sound NIZK proof system for $L_{g,f,h}$.*

Again, it is worth pointing out here that we use the fact that in Groth-Sahai NIZKs for DLIN, once the commitments to the witnesses are fixed, there is a unique proof satisfying the linear equations of the type used in the above NIZK proof.

Again, it is easy to extend this proof system to a labeled version in the following way. Compute t as before, but additionally include the label. That is, compute t as $H(\rho, \sigma, \tau, com, label)$.

F.2 Public Verifiable CCA2 Encryption

We now define a *labeled* publicly-verifiable public-key encryption scheme DLENC as follows:

Key Generation: Generate $g, f, h \stackrel{\$}{\leftarrow} \mathcal{G}$, and $k_1, k_2 \stackrel{\$}{\leftarrow} \mathbb{Z}_q^*$. Let $K_1 = g^{k_1}$ and $K_2 = f^{k_2}$. Let ψ be the CRS for a Labeled 1-SRS-NIZK for the language $L_{g,f,h}$. The public key is $(g, f, h, K_1, K_2, \psi)$ and the private key is (k_1, k_2).

Encrypt: Given plaintext $m \in \mathcal{G}$, and label $\mathtt{l}$. Choose $x, y \stackrel{\$}{\leftarrow} \mathbb{Z}_q^*$. Let the tuple $\langle \rho, \sigma, \tau, \gamma \rangle$ be $\langle g^x, f^y, h^{x+y}, m \cdot {K_1}^x {K_2}^y \rangle$. Let π be a 1-SRS-NIZK proof of $(\rho, \sigma, \tau) \in L_{g,f,h}$ with witness (x, y) and label $(\gamma, \mathtt{l})$. The ciphertext is $(\rho, \sigma, \tau, \gamma, \pi)$.

Decrypt: Given ciphertext $c = (\rho, \sigma, \tau, \gamma, \pi)$ and label $\mathtt{l}$. Verify if π is a 1-SRS-NIZK proof for (ρ, σ, τ) and label $(\gamma, \mathtt{l})$. If verification fails output $\perp$. Otherwise output $m = \frac{\gamma}{\rho^{k_1} \sigma^{k_2}}$.

Verify: Given ciphertext $c = (\rho, \sigma, \tau, \gamma, \pi)$ and label $\mathtt{l}$. Verify if π is a 1-SRS-NIZK proof for (ρ, σ, τ) and label $(\gamma, \mathtt{l})$. If verification fails output false else output true.

Theorem 8 *The scheme* DLENC *is publicly-verifiable (labeled) IND-CCA2 secure under the DLIN assumption.*

F.3 Secure Protocol in the PAK Model

We again instantiate the [19] scheme, but now under the DLIN assumption. The public verifiable encryption is the scheme DLENC as described before. Let the public key for the DLENC scheme be $(g, f, h, K_1, K_2, \psi)$. The hash proof system is described as follows:

Key Generation: Generate $l, m, n \xleftarrow{\$} \mathbb{Z}_q^*$. Compute $\eta \leftarrow g^l (K_1)^n$ and $\phi \leftarrow f^m (K_2)^n$. The public key is (η, ϕ) and the private key is (l, m, n).

Public Hash Computation: Given parameters $(label, c, msg)$, where $c = \langle \rho, \sigma, \tau, \gamma, \pi \rangle$. Also given $(\rho, \sigma, \tau; x, y) \in \mathcal{R}_{f,g,h}$. Then compute hash as:

$$H_{\eta,\phi}(label, c, msg) \leftarrow \eta^x \phi^y$$

Private Hash Computation: Given parameters $(label, c, msg)$, where $c = \langle \rho, \sigma, \tau, \gamma, \pi \rangle$. Then compute hash as:

$$H_{l,m,n}(label, c, msg) \leftarrow \rho^l \sigma^m \left(\frac{\gamma}{msg} \right)^n$$

Theorem 9 *Assume the existence of a DLIN hard group G which supports a bilinear pairing operation. Then the protocol in Figure 1 with encryption instantiated by* DLENC *and hash proof system instantiated as described, is secure in the PAK model.*

F.4 Secure PWKE-Protocol in the UC/DLIN Model

In Figure 4, we give a UC-secure PWKE-protocol under the Decisional Linear assumption (DLIN).

Theorem 10 *Assume the existence of a DLIN-hard group, a labeled unbounded simulation-sound G-extractable NIZK proof system. Then the protocol in Figure 4 securely realizes the* $\widehat{\mathcal{F}}_{\text{PWKE}}$ *functionality in the* $\mathcal{F}_{\mathsf{crs}}$ *hybrid model, in the presence of static corruption adversaries.*

UC Protocol for Password-based Key Exchange in DLIN Group G

$\text{CRS} = g, f, h, K_1, K_2, \mathcal{P}, \psi : \quad g, f, h, K_1, K_2 \xleftarrow{\$} G \quad \mathcal{P} \xleftarrow{\$} G \quad \psi = \text{uSS-NIZK } \mathsf{CRS}$

Party P_i		Adversary $\mathcal{A}$
Input $(\texttt{NewSession}, sid, \mathsf{ssid}, P_i, P_j, \text{pwd}, initiator/responder)$		
Choose $x_i, y_i, l_i, m_i, n_i \xleftarrow{\$} \mathbb{Z}_q^*$.		
Set $\begin{bmatrix} \rho_i = g^{x_i}, \sigma_i = f^{y_i}, \tau = h^{x_i+y_i}, \\ \gamma_i = \text{pwd} \cdot (K_1^{x_i} K_2^{y_i})^{\mathsf{ssid}}, \\ \eta_i = g^{l_i} K_1^{n_i}, \phi_i = f^{m_i} K_2^{n_i} \end{bmatrix}$	$\xrightarrow{c_i, \eta_i, \phi_i, \pi_i}$	$\mathcal{A}$
Let $c_i = \langle \rho_i, \sigma_i, \tau_i, \gamma_i \rangle$, and $\pi_i = \text{uSS-NIZK}_\psi \left(\begin{array}{c} \rho_i, \sigma_i, \tau_i, \eta_i, \phi_i; \\ x_i, y_i, \mathcal{P}^{l_i}, \mathcal{P}^{m_i}, \mathcal{P}^{n_i} \end{array} \right)$ with label $\langle P_i, P_j, \mathsf{ssid} \rangle$.		
	$\xleftarrow{c'_j, \eta'_j, \phi'_j, \pi'_j}$	$\mathcal{A}$
Let $c'_j = \langle \rho'_j, \sigma'_j, \tau'_j, \gamma'_j \rangle$.		
If any of $\rho'_j, \sigma'_j, \tau'_j, \gamma'_j, \eta'_j, \phi'_j$ is not in $G \backslash \{1\}$, or not uSS-NIZK-Verify$(\pi'_j; \rho'_j, \sigma'_j, \tau'_j, \eta'_j, \phi'_j)$ with label $\langle P_j, P_i, \mathsf{ssid} \rangle$		
Set $\text{sk}_i \xleftarrow{\$} G_T$,		
else compute $\begin{bmatrix} h'_2 = (\rho'_j)^{l_i} (\sigma'_j)^{m_i} (\frac{\gamma'_j}{\text{pwd}})^{n_i} \\ h_1 = (\eta'_j)^{x_i} (\phi'_j)^{y_i} \end{bmatrix}$		
Set $\text{sk}_i = e(h'_2 \cdot h_1, \mathcal{P})$.		
Output $(sid, \mathsf{ssid}, \text{sk}_i)$.		

Figure 4: Single round UC-secure Password-based Authenticated Key Exchange in DLIN Bilinear group G of prime order q. It assumes a group G with a $\mathbb{Z}_q^*$-bilinear map e from $G \times G$ to G_T. Let g and $\mathcal{P}$ be generators of G. The shared password pwd is assumed to be in G, and ssid is assumed to be in $\mathbb{Z}_q^*$. The uSS-NIZK$(\rho, \sigma, \tau, \eta, \phi;\ x, y, L, M, N)$ is a labeled unbounded-simulation G-extractable NIZK proof of membership in the language $L = \{\rho, \sigma, \tau, \eta, \phi \mid \exists x, y, L, M, N : \rho = g^x, \sigma = f^y, \tau = h^{x+y}, e(\eta, \mathcal{P}) = e(g, L) e(K_1, N), e(\phi, \mathcal{P}) = e(f, M) e(K_2, N)\ \}$