Relatively-Sound NIZKs and Password-Based Key-Exchange*

Charanjit Jutla[1] and Arnab Roy[2]

[1] IBM T.J. Watson Research Center,
Yorktown Heights, NY 10598, USA
[2] Fujitsu Laboratories of America,
Santa Clara, CA 94058, USA

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.

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 resulting in insufficient randomness for generating cryptographically secure keys. Gong et al. [11] raised the problem of designing protocols resistant to offline password guessing attacks, where other than guessing the low-entropy password by an online attack, the protocol must otherwise provide strong security based on a security parameter. Beginning with the work of Bellovin and Merritt [2], there has been considerable theoretical work in formalizing and obtaining secure protocols in the setting where only passwords are shared by peers (e.g. [1]), referred to as the

* Authors were supported in part by the Department of Homeland Security under grant FA8750-08-2-0091.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 485–503, 2012.

PAK-security model. From [15] onwards, 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 [7].

As illustrated by Gennaro and Lindell [10], 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 [10] requires an adaptive smooth hash key, which makes the key-exchange protocol a multi-round protocol.

Recently, Katz and Vaikuntanathan [16] gave a single round protocol for password-based authenticated key exchange, by utilizing a publicly-verifiable CCA2-encryption scheme of Sahai [19]. 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. The public verification allows the smooth hash to be defined on only a part of the ciphertext, which in [16] happens to be two El-Gamal encryptions of the password. Such smooth projective hashes are easy to define and compute.

While the resulting protocol requires only a constant number of group elements, as it employs simulation-sound extensions of Groth-Sahai NIZKs [13], under the decisional linear assumption (DLIN [3]) it still requires each party to send 65 group elements (and the run-time is proportionately high).

In this paper we show that the above scheme can be made much more efficient by using a novel concept of *relatively-sound* NIZKs rather than using simulation-sound NIZKs. Simulation-Sound NIZKs were first defined by Sahai [19], where it was used to convert Naor-Yung [18] CCA1-encryption scheme into the aforementioned 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. The novel relatively-sound NIZKs captures this idea[1]. While it is an open problem whether relatively-sound NIZKs are strictly weaker than adaptive simulation-sound NIZKs, we show that relatively-sound NIZKs imply soundness under simulation of proofs of random (false or true)

[1] Relatively-sound NIZKs can be considered a hybrid of designated-verifier simulation-sound NIZKs [9] and simulation-sound NIZKs.

statements. Since, for many applications (including the current) such non-adaptive (random) simulation-sound NIZKs suffice, relative-soundness can be seen as a useful abstraction and tool for obtaining the former.

While it is easy to check that relative-soundness suffices in Sahai's original proof, in this paper we consider a further optimized construction. We prove that an augmented El-Gamal encryption scheme (reminiscent of [8]), 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 [7], which allows us to build a private verifier. Under the SXDH assumption [13], 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 [16]. Overall under the DLIN assumption, our publicly-verifiable CCA2 ciphertexts have only 19 group elements versus the 47 group elements in the Sahai scheme [19].

We show that using the new encryption scheme in the PAK-model protocol of [16], 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[2].

UC Security. Canetti et al. [6] proposed a definition of security for password-based key exchange protocols within the Universally Composable (UC) security framework [5], 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 [16] 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 in their protocol makes it rather expensive.

[2] It should be remarked that other efficient publicly-verifiable CCA2-encryption schemes such as [17], which allow hash proofs on the (proof-less) part of the ciphertext can also be used in [16].

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 [4] which is based on a construction due to Groth [12], 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 [8] 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 [16]. 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 full paper [14].

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. [19,13]). 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 poly-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
$\Pr[(\sigma, \tau) \leftarrow S_1(1^m); (x, s) \leftarrow \mathcal{A}_1(\sigma); \pi \leftarrow S_2(\sigma, \tau, x); (x', \pi') \leftarrow \mathcal{A}_2(x, \pi, \sigma, s) :$
$((x', \pi') \neq (x, \pi))$ and $\neg\exists w'$ s.t. $(x', w') \in R$, and $V(\sigma, x', \pi') = 1] \approx 0$.

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

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 simulator's private verifier, and hence the name *relative-soundness*. There is an important further stipulation in the definition that the zero-knowledge property should hold even when the Adversary is given oracle access to private verifier in the simulated world (and public verifier in real world).

Labeled Single-Theorem Relatively-Sound NIZK (l-SRS-NIZK). Consider a sound and complete (labeled) proof (K, P, V) for a relation R along with a PPT private-verifier W and a PPT simulator (S_1, S_2). In a labeled proof, the prover P takes an input label, in addition to the statement to be proven. The verifier takes a statement, a label, and a proof. Such a proof is called a **labeled 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

relative-ZK:

$$\Pr[(\psi) \leftarrow K(1^m);\ \ (x, w, \mathtt{lbl}, s) \leftarrow \mathcal{A}_1^{V(\psi,\cdot,\cdot,\cdot)}(\psi);\ \ \pi \leftarrow P(\psi, x, w, \mathtt{lbl}) : \mathcal{A}_2^{V(\psi,\cdot,\cdot,\cdot)}(\pi, s) = 1] \approx$$

$$\Pr[(\sigma, \tau) \leftarrow \mathrm{S}_1(1^m);\ \ (x, w, \mathtt{lbl}, s) \leftarrow \mathcal{A}_1^{W(\sigma,\tau,\cdot,\cdot,\cdot)}(\sigma);\ \ \pi \leftarrow S_2(\sigma, \tau, x, \mathtt{lbl}) : \mathcal{A}_2^{W(\sigma,\tau,\cdot,\cdot,\cdot)}(\pi, s) = 1],$$

for A_1 restricted to producing (x, w) satisfying R, **and**

relative-simulation-soundness:

$$\Pr[(\sigma, \tau) \leftarrow \mathrm{S}_1(1^m); (x, \mathtt{lbl}, s) \leftarrow \mathcal{A}_3^{W(\sigma,\tau,\cdot,\cdot,\cdot)}(\sigma); \pi \leftarrow S_2(\sigma, \tau, x, \mathtt{lbl}); (x', \mathtt{lbl}', \pi') \leftarrow \mathcal{A}_4^{W(\sigma,\tau,\cdot,\cdot,\cdot)}(\pi, s) : \ \ ((x', \mathtt{lbl}', \pi') \neq (x, \mathtt{lbl}, \pi)) \text{ and } \neg\exists w' \text{ s.t. } R(x', w') = 1, \text{ and } W(\sigma, \tau, x', \mathtt{lbl}', \pi') = 1] \approx 0.$$

Note that there are no other requirements on W other that those listed above. It is critical that relative-ZK is required only w.r.t. adversaries $(\mathcal{A}_1)$ that produce language members. Otherwise, relative-simulation-soundness would already imply normal simulation-soundness. Although it remains an open problem whether relatively-sound NIZKs are *strictly* weaker than simulation-sound NIZKS, the following shows the relation to non-adaptive simulation soundness, i.e. where the statements for which the proofs need to be simulated are chosen randomly.

Relation to Simulation-Soundness. Consider the following variant of One-time Simulation Soundness defined in Section 2. A NIZK proof for language

$L \subseteq X$ is a **non-adaptive one-time simulation-sound NIZK** if for all non-uniform PPT adversaries $\mathcal{A} = (\mathcal{A}_3, \mathcal{A}_4)$ we have

$\Pr[(\sigma, \tau) \leftarrow \mathrm{S}_1(1^m); x \xleftarrow{\$} X; (\mathtt{lbl}, s) \leftarrow \mathcal{A}_3(\sigma, x); \pi \leftarrow S_2(\sigma, \tau, x, \mathtt{lbl});$
$(x', \mathtt{lbl}', \pi') \leftarrow \mathcal{A}_4(\pi, s) : ((x', \mathtt{lbl}', \pi') \neq (x, \mathtt{lbl}, \pi))$
and $\neg\exists w'$ s.t. $R(x', w') = 1$, and $V(\sigma, x', \mathtt{lbl}', \pi') = 1] \approx 0$.

Now, assume that the language L is *efficiently witness-samplable*, i.e. there is PPT machine which can efficiently sample from L along with the witness for the language member. Also, a language L, subset of a domain X, is called *hard* if no PPT adversary can distinguish between a (uniformly) random element of L from a random element of X.

Lemma 1. *For a hard and efficiently witness-samplable language L, an l-SRS-NIZK for L also satisfies the non-adaptive labeled one-time simulation soundness property for L.*

The proof of this lemma uses standard arguments, and a version of this lemma for unbounded simulation soundness also holds.

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 as follows. For K, m in G, and x, define

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

For K and pwd in G, define $L_{K,\mathrm{pwd}} = \{c = \langle R, P \rangle \mid \exists x : c = \mathsf{enc}^{\mathsf{eg}}_K(\mathrm{pwd}; x)\} \cap G \times G$. A **projective hash function** [7] 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** $\alpha : \mathcal{K} \to S$, where $\mathcal{K}$ 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

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

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

For any K and pwd in G, $\mathcal{H}^{K,\mathrm{pwd}}$ is said to be **smooth** [7] 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 $(\mathcal{H}^{K,\mathrm{pwd}}_{n,\hat{n}}(c'),\ \alpha^{K,\mathrm{pwd}}(n.\hat{n}))$ and the pair (g^{d_1}, g^{d_2}) is negligible, where $n, \hat{n}, d_1, d_2$ are chosen randomly and independently from $\mathbb{Z}_q$. 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 any language L to be **2-universal** [7] 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 q, which allow an efficiently computable $\mathbb{Z}_q$-bilinear pairing map $e : G_1 \times G_2 \to G_T$.

SXDH: [13] 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.

DLIN: [3] 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 $\alpha, \beta, r, s \in \mathbb{Z}_q$, and arbitrary generator $\mathcal{P}$ of G_1, it is hard to distinguish between $t = r + s$ and a random t.

5 A Publicly-Verifiable CCA2-Encryption Scheme

In this section we describe a CCA2-Encryption scheme that has the property that a potential ciphertext can be publicly verified to be a valid ciphertext of some message. Note that Sahai [19] had previously given a publicly-verifiable CCA2-encryption scheme employing the Naor-Yung CCA1-scheme [18], 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 [7] was correctly computed. However, since the NIZK scheme by itself may be 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 [19]), 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, A^x. 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 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 $g, A \xleftarrow{\$} G_1$, and $k \xleftarrow{\$} \mathbb{Z}_q$. Let $K = g^k$. Let ψ be the CRS for an l-SRS-NIZK. The public key is (g, A, K, ψ) and the private key is k.

Encrypt: Given plaintext $m \in G_1$, and label $\texttt{lbl}$. 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 an l-SRS-NIZK proof of $((\rho, \hat{\rho}), x) \in \mathcal{R}$ with label $\gamma, \texttt{lbl}$. The ciphertext is $(\rho, \hat{\rho}, \gamma, \pi)$.

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

Verify: Given ciphertext $c = (\rho, \hat{\rho}, \gamma, \pi)$ and label $\texttt{lbl}$. Verify if π is an l-SRS-NIZK proof for $(\rho, \hat{\rho})$ and label $\gamma, \texttt{lbl}$. 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 [14], 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, and the decryption oracle in the original experiment does a public verification of proofs in each adversarially supplied ciphertext. (2) The NIZK CRS is switched to be the hiding CRS, the proof switched to a simulator generated proof, and decryption oracle now uses private-verification. This is an indistinguishable change by the relative-ZK property of l-SRS-NIZK. Note, x is no more used in the simulated proof. (3) The decryption is done as $\gamma/(\rho^{k'}\hat{\rho}^{k''})$, which is equivalent because of relative-simulation soundness property of l-SRS-NIZK. (4) DDH is employed, as only $A(= g^a)$ is being used in the simulation, instead of a. This leads to A^x being replaced by an independent X'. (5) The decryption is done as γ/ρ^k, which is again equivalent by relative-soundness. (6) the message in the encryption can be switched by pairwise independence in k, and this step is information-theoretic. More precisely, $g^{xk'}(X')^{k''}$ is random and independent of g^x, X', K, A, as well as Adversary's coins with high probability. (7) Next we do all the above steps (2)-(5) in reverse.

6 l-SRS-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 $\texttt{SXDH}$ 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 [7]. In [7], Cramer and Shoup show how to obtain 2-universal projective hash functions for such languages, and we use these hash functions for private-verification.

We construct an l-SRS-NIZK proof system for $L_{g,A}$, as follows:

CRS Generation: Generate $\mathcal{P} \xleftarrow{\$} G_2$ and $u, v, d_1, d_2, e_1, e_2 \xleftarrow{\$} \mathbb{Z}_q$. 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 (*binding* CRS), and the last two are the projection keys for a 2-universal projective-hash for the DDH language (just as [7]), to be used in the relatively-sound system.
The simulation CRS σ is $(P, Q, R, S, \mathbf{d}, \mathbf{e}) = (\mathcal{P}, \mathcal{P}^u, \mathcal{P}^v, \mathcal{P}^{uv}, g^{d_1}A^{d_2}, g^{e_1}A^{e_2})$. This is the *hiding* CRS of GS-NIZK for SXDH along with $\mathbf{d}$ and $\mathbf{e}$ as above. The trapdoor is $\tau = (u, d_1, d_2, e_1, e_2)$.

Prover: Given witness x, candidate (g^x, A^x), and label $\texttt{lbl}$, construct proof as follows. Generate $s \xleftarrow{\$} \mathbb{Z}_q$. Compute $t \leftarrow H(g^x, A^x, Q^x P^s, S^x R^s, \texttt{lbl})$, where H is a collision resistant hash function. Then compute: $(\beta, c_1, c_2, \theta, \phi, \chi) \leftarrow ((\mathbf{de}^t)^x, Q^x P^s, S^x R^s, g^s, A^s, (\mathbf{de}^t)^s)$. 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, $\hat{\rho}$, $\texttt{lbl}$, *and the commitment* to x in the NIWI system, i.e. $Q^x P^s, S^x R^s$.

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

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

Public Verify: Given $\pi = (\beta, c_1, c_2, \theta, \phi, \chi)$ as a candidate proof of $(\rho, \hat{\rho})$ with label $\texttt{lbl}$, compute $t \leftarrow H(\rho, \hat{\rho}, c_1, c_2, \texttt{lbl})$. 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 $(\rho, \hat{\rho})$ with label $\texttt{lbl}$, compute $t \leftarrow H(\rho, \hat{\rho}, c_1, c_2, \texttt{lbl})$. Then first do public verification and if that succeeds then check the following equation: $\beta \stackrel{?}{=} \rho^{d_1}\hat{\rho}^{d_2}(\rho^{e_1}\hat{\rho}^{e_2})^t$. Note that this private verifier is well-defined in the real world as well. In addition, its trapdoor (d_1, d_2, e_1, e_2) is identically generated in both the real and the simulated worlds.

Theorem 2. *The above system is an l-SRS-NIZK proof system for* $L_{g,A}$.

Proof Sketch: We focus on Relative-ZK and Relative-SS properties. For the former, we need to show that the simulation CRS, and a proof for $(\rho, \hat{\rho})$ with label $\texttt{lbl}$ is computationally indistinguishable from the real CRS and a real proof, even when the Adversary has oracle access to respective verifiers. This is accomplished by a sequence of games, where the first game is same as the real world game. In the second game, the CRS and the proof remain the same but the verifier in the oracle is changed to be the private verifier, which in our case is well-defined in the real world. We need to show that public verification implies

private verification, but this follows from soundness of the Groth-Sahai NIZK, as well as the fact that on a *valid* DDH tuple the projection hash is same whether it is computed using the witness and the projection key or using the private hash keys. In the final game we switch to the simulation CRS and simulated proof, and indistinguishability follows from ZK property of Groth-Sahai NIZKs and the fact that the private verification trapdoor is independent of the Groth-Sahai NIZK CRS (hiding or binding).

The relative-simulation-soundness property is proven using the 2-universal property of the projective smooth hash (just as in [7]), but additionally using 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. □

The l-SRS-NIZK proof for DDH language above consists of six group elements. The l-SRS-NIZK proof for the DLIN language (and under the DLIN assumption), given in the full paper [14], 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 [1]. We instantiate the single-round scheme due to Katz and Vaikuntanathan [16], which is described in Figure 1, with the more efficient publicly-verifiable CCA-secure encryption scheme DHENC of Section 5, which enables a more efficient hash proof as well. The common reference string (CRS) is just the public key of this scheme.

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

$$\mathrm{CRS} = pk$$

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

Fig. 1. Single-round PAK-Model Secure Password-based Authenticated KE

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 [16], 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 $L_{K,\mathrm{pw}}$, which are CPA encryptions of password.

8 Secure Protocol in the UC Model

The essential elements of the Universal Composability framework can be found in [5]. We adopt the definition for password-based key exchange from Canetti et al [6]. The following description is a summary from [6]. The formal description is given in Figure 2.

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

The functionality $\mathcal{F}_{\mathrm{pwKE}}$ 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** 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.

Fig. 2. The password-based key-exchange functionality $\mathcal{F}_{\mathrm{pwKE}}$

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. 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, and the Adversary gets the power to set the session key.

8.1 A Single Round UC Password-Based Key Exchange Protocol

The single-round UC protocol under the SXDH assumption uses labeled unbounded simulation sound G_2-extractable NIZKs (uSS-NIZK). 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. Since g, $\mathcal{P}$ 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 $x \xleftarrow{\$} \mathbb{Z}_q$ and computes $c_1 = \langle g^x, A^x, K^x \cdot pw\rangle$. It also generates hash key $(n_1, \hat{n_1}) \xleftarrow{\$} (\mathbb{Z}_q)^2$ 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:

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

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 3 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.

A more optimized version of such a general labeled unbounded simulation sound G_2-extractable NIZK [7] is given in the Appendix in Section A. In fact,

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

Party P_i		Adv $\mathcal{A}$
Input (NewSession, sid, ssid, P_i, P_j, 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 $\langle P_i, P_j, \text{ssid} \rangle$.		
	$\xleftarrow{c_2', \eta_2', \pi_2'}$	$\mathcal{A}$
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 $\langle P_j, P_i, \text{ssid} \rangle$		
set $\text{sk}_1 \xleftarrow{\$} G_T$, else		
compute $h_2' = (\frac{\gamma_2'}{\text{pwd}})^{\hat{n}_1}(\rho_2')^{n_1}$, $h_1 = (\eta_2')^{x_1}$, $\text{sk}_1 = e(h_2' \cdot h_1, \mathcal{P})$.		
Output $(sid, \text{ssid}, \text{sk}_1)$.		

Fig. 3. Single round UC-secure Password-based KE under SXDH Assumption

for the language above for which such a NIZK is required, we give a further optimization in [14]. 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 (given in the full paper [14]) requires 54 group elements.

8.2 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 $\mathcal{Z}$, the functionality $\widehat{\mathcal{F}}_{\text{PWKE}}$, and uses $\mathcal{A}$ as a subroutine. The messages between $\mathcal{Z}$ 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.

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.

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 $\mathcal{Z}$, including $\mathcal{F}$'s communication with $\mathcal{Z}$):

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 extraction trapdoor ξ.
5. Compute $\text{pwd}' = \gamma'_2/(\rho'_2)^k$. If (pwd′ ≠ 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 $e(\mu/\text{pwd}, \hat{N}'_2)$.

8.3 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 8.2 above, and ending with an experiment which is identical to the real world execution of the protocol in Fig 3. We will show that the environment has negligible advantage in distinguishing between these experiments, leading to a proof of realization of $\mathcal{F}_{\text{PWKE}}$ by the protocol Π.

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 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

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

Because the hash proof system is for languages with messages encrypting real password, the smooth-hash-proof yields random values from the adversary's point of view. Note that 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 the 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, just as in CCA2 scheme DHENC described in Section 5.

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, and further we drop 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 $e(h'_2, \mathcal{P}) \cdot e(\eta_2'^{x_1}, \mathcal{P})$. We replace all simulator generated proofs by proofs generated by real prover, and switch from the CRS generated by SE_1 to the real world CRS. Experiment Expt_6 is indistinguishable from the real-world experiment by completeness of the hash proof system, i.e. when the labeled tuple c, ssid is in the language, then the hash can be computed from the projection keys and the witness x_1 of c. This completes the proof of Theorem 4. □

Acknowledgments. The authors would like to thank the referees for several helpful comments.

References

1. Bellare, M., Pointcheval, D., Rogaway, P.: Authenticated Key Exchange Secure against Dictionary Attacks. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 139–155. Springer, Heidelberg (2000)
2. Bellovin, S.M., Merritt, M.: Encrypted key exchange: Password-based protocols secure against dictionary attacks. In: 1992 IEEE Symposium on Security and Privacy, pp. 72–84. IEEE Comp. Soc. Press (May 1992)
3. Boneh, D., Boyen, X., Shacham, H.: Short Group Signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
4. Camenisch, J., Chandran, N., Shoup, V.: A Public Key Encryption Scheme Secure against Key Dependent Chosen Plaintext and Adaptive Chosen Ciphertext Attacks. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 351–368. Springer, Heidelberg (2009)
5. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: 42nd FOCS, pp. 136–145. IEEE Comp. Soc. Press (October 2001)
6. Canetti, R., Halevi, S., Katz, J., Lindell, Y., MacKenzie, P.: Universally Composable Password-Based Key Exchange. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 404–421. Springer, Heidelberg (2005)
7. Cramer, R., Shoup, V.: Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002)
8. Damgård, I.B.: Towards Practical Public Key Systems Secure against Chosen Ciphertext Attacks. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 445–456. Springer, Heidelberg (1992)

9. Elkind, E., Sahai, A.: A unified methodology for constructing public-key encryption schemes secure against adaptive chosen-ciphertext attacks. Cryptology ePrint Archive: Report 2002/042
10. Gennaro, R., Lindell, Y.: A Framework for Password-based Authenticated Key Exchange. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 524–543. Springer, Heidelberg (2003)
11. Gong, L., Lomas, T.M.A., Needham, R.M., Saltzer, J.H.: Protecting poorly chosen secrets from guessing attacks. IEEE JSAC 11(5), 648–656 (1993)
12. Groth, J.: Simulation-Sound NIZK Proofs for a Practical Language and Constant Size Group Signatures. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 444–459. Springer, Heidelberg (2006)
13. Groth, J., Sahai, A.: Efficient Non-interactive Proof Systems for Bilinear Groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008)
14. Jutla, C., Roy, A.: Relatively-sound NIZKs and password-based key-exchange. Cryptology ePrint Archive, Report 2011/507 (2011), `http://eprint.iacr.org/`
15. Katz, J., Ostrovsky, R., Yung, M.: Efficient Password-Authenticated Key Exchange Using Human-Memorable Passwords. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 475–494. Springer, Heidelberg (2001)
16. Katz, J., Vaikuntanathan, V.: Round-Optimal Password-Based Authenticated Key Exchange. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 293–310. Springer, Heidelberg (2011)
17. Kiltz, E.: Chosen-Ciphertext Security from Tag-Based Encryption. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 581–600. Springer, Heidelberg (2006)
18. Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: 22nd ACM STOC. ACM Press (May 1990)
19. Sahai, A.: Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In: 40th FOCS, pp. 543–553. IEEE Comp. Soc. Press (1999)

A More Efficient Unbounded Simulation Sound NIZKs

In [4], 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 [4] 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 [4] 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 groups (G_1, G_2, G_T), with a $\mathbb{Z}_q$-bilinear map e. 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 $\boldsymbol{x} \cdot \boldsymbol{A} = T$, where $\boldsymbol{x}$ are variables from $\mathbb{Z}_q$, $\boldsymbol{A}$ are constants from G_2, and T is a constant from G_T, and thus $\boldsymbol{x}$ serves as witness for a member of a language specified by $\boldsymbol{A}$ and T. Languages can also be specified by equations of the form $\boldsymbol{B} \cdot \boldsymbol{\mathcal{Y}} = T_1 \cdot T_2$, where $\boldsymbol{B}$ are elements from G_1, $\boldsymbol{\mathcal{Y}}$ 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 $\boldsymbol{x} \cdot \boldsymbol{A} + \boldsymbol{b} \cdot \boldsymbol{\mathcal{Y}} = T$, as well as quadratic equations.

The uss-NIZK CRS will consist of the usual Groth-Sahai NIWI CRS for SXDH, along with $g, A{=}g^a, \mathbf{k} = g^{k_1} A^{k_2}, \mathbf{d}{=}g^{d_1} A^{d_2}, \mathbf{e}{=}g^{e_1} A^{e_2}$, and $\mathbf{h}{=}g^x, \mathbf{u}{=}g^u$, with $g \in G_1$, and $a, k_1, k_2, d_1, d_2, e_1, e_2, x, u$ chosen at random from $\mathbb{Z}_q$. 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 $\boldsymbol{x} \cdot \boldsymbol{A} = T$, it generates a modified equation $\boldsymbol{x} \cdot \boldsymbol{A} = \delta \cdot T$, where δ is a new global integer variable.
   – Get modified equations of the form $\boldsymbol{B} \cdot \boldsymbol{\mathcal{Y}} + T_1 \cdot \mathcal{V} = 0$, where $\mathcal{V}$ is a new variable representing elements from G_2, along with an additional equation $\mathcal{V} + (\delta - 1) \cdot T_2 = 0$ [13].
   – 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 $b = \mathbf{u} \cdot (\mathbf{k}\mathbf{d}\mathbf{e}^t)^w$, 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): $b^{1-\delta} = \mathbf{u}^{x'} \cdot (\mathbf{k}\mathbf{d}\mathbf{e}^t)^{w'}$. Call this proof π_3.
7. The uss-NIZK proof consists of $(\pi_1, \pi_2, \pi_3, \rho, \hat{\rho}, b)$.

The proof of zero-knowledge is similar to the construction in [4]. The proof of unbounded simulation sound extractability is also similar to as in [4] but using the CCA2 encryption scheme (and its proof) as described in Section 5.

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.

Note that it takes 14 extra group elements to convert an SXDH based NIZK proof into a uSS-proof using this construction (and 28 elements for a DLIN based construction) [13]. For the language in Section 8.1, the NIZK proof requires 18 group elements. In the full paper [14] we show a further optimization for this specific language, which saves another 3 group elements, resulting in a total of 29 group elements for a uss-NIZK proof for the language.