The Group of Signed Quadratic Residues and Applications

Dennis Hofheinz* and Eike Kiltz**

Abstract. We consider the cryptographic group of *Signed Quadratic Residues*. This group is particularly useful for cryptography since it is a "gap-group," in which the computational problem (i.e., computing square roots) is as hard as factoring, while the corresponding decisional problem (i.e., recognizing *signed* quadratic residues) is easy. We are able to show that under the factoring assumption, the Strong Diffie-Hellman assumption over the signed quadratic residues holds. That is, in this group the Diffie-Hellman problem is hard, even in the presence of a Decisional Diffie-Hellman oracle.

We demonstrate the usefulness of our results by applying them to the Hybrid ElGamal encryption scheme (aka Diffie-Hellman integrated encryption scheme — DHIES). Concretely, we consider the security of the scheme when instantiated over the group of signed quadratic residues. It is known that, in the random oracle model, the scheme is chosen-ciphertext (CCA) secure under the Strong Diffie-Hellman assumption and hence, by our results, under the standard factoring assumption. We show that furthermore, in the standard model, Hybrid ElGamal is CCA secure under the higher residuosity assumption, given that the used hash function is four-wise independent. The latter result is obtained using the recent "randomness extraction framework" for hash proof systems.

Keywords: Public-key encryption, chosen-ciphertext security, Hybrid ElGamal/DHIES.

1 Introduction

1.1 Quadratic Residues

The group of quadratic residues $\mathbb{QR}_N$ over a Blum integer $N = PQ$ (where $P \equiv Q \equiv 3 \bmod 4$) has proven to be a useful group for cryptographic purposes. For example, Rabin [30] proved that computing square roots in this group is equivalent to factoring the modulus N. The latter is believed to be hard in general ("factoring assumption"). Rabin's fundamental observation is the basis for a number of cryptographic protocols that are provably secure under the factoring assumption (e.g., the encryption and signature schemes [30,5,21]).

The quadratic residues have yet another useful property. Namely, given a uniformly random element modulo N (with Jacobi symbol 1), it is believed to be hard to decide whether the element is a square or not. This is the *quadratic*

* CWI, Amsterdam, Dennis.Hofheinz@cwi.nl. Work supported by NWO.
** CWI, Amsterdam, kiltz@cwi.nl. Work supported by NWO.

S. Halevi (Ed.): CRYPTO 2009, LNCS 5677, pp. 637–653, 2009.

residuosity assumption, a stronger assumption than the factoring assumption. On the bright side, there are again numerous cryptographic protocols whose security relies on the quadratic residuosity assumption (e.g., [18,11]).

However, the quadratic residuosity assumption also has a dark side. Namely, whenever an active adversary may choose group elements as protocol inputs (such as ciphertexts submitted for decryption), the receiving (honest) party may not be able to distinguish quadratic residues from quadratic non-residues. In particular, the adversary may learn some secret information by observing the protocol's different behaviour on quadratic residues and non-residues. Concretely, this problem naturally occurs when trying to reduce the chosen-ciphertext security (CCA security) of an encryption scheme (defined over the quadratic residues) to the factoring assumption. Specifically, during such a reduction, a decryption oracle has to be implemented without the knowledge of the factorization of N. Hence, the decryption oracle cannot distinguish quadratic residues from non-residues. This allows an adversary that uses the decryption oracle to submit, say, both $C \in \mathbb{Z}_N^*$ and $-C \in \mathbb{Z}_N^*$ (one of which is not a square) for decryption. This makes implementing a decryption oracle harder, in particular since the the submitted non-squares could be related to the challenge ciphertext.

Another intractability problem commonly used in cryptography is the Diffie-Hellman (DH) problem [13]. Given a generator g of a cyclic group $\mathbb{G}$ and $X = g^x, Y = g^y$, the DH key is defined as $\mathrm{DH}_g(X, Y) = g^{xy}$. The (Computational) DH problem is to compute $\mathrm{DH}_g(X, Y)$ from g, X, Y. For passive (chosen-plaintext) adversaries the security of the DH key exchange protocol [13] and the ElGamal encryption scheme [15] is equivalent to the DH problem. Over the group of quadratic residues (i.e., if $\mathbb{G} = \mathbb{QR}_N$), Shmuely [32] and McCurley [27] proved that the DH problem is at least as hard as factoring N.

The Strong Diffie-Hellman (SDH) problem [1] is to compute $\mathrm{DH}_g(X, Y)$ from g, X, Y while having access to a (Decisional) DH oracle that returns 1 on input $(\hat{Y}, \hat{Z})$ if $\mathrm{DH}_g(X, \hat{Y}) = \hat{Z}$ and $(\hat{Y}, \hat{Z}) \in \mathbb{G} \times \mathbb{G}$ (and 0 otherwise). Interestingly, for active (chosen-ciphertext) adversaries, the security of the (hashed) Diffie-Hellman key exchange protocol [13] and the Hybrid ElGamal encryption scheme [15] is equivalent to the SDH problem [9] in the random oracle model [3]. However, the result of Shmuely does not extend to prove that the SDH problem is at least as hard as factoring, since to simulate the DH oracle, one must be able to determine membership in the quadratic residues.

1.2 Signed Quadratic Residues

We propose to use a cryptographic group we call the *Signed Quadratic Residues* ($\mathbb{QR}_N^+$). This group has been suggested already by Fischlin and Schnorr in [16, Section 6] (in the different context of hard-core bits for generalized Rabin functions), but has not been investigated any further. This group is useful for cryptography since membership in $\mathbb{QR}_N^+$ can be publicly (and efficiently) verified while it inherits some nice intractability properties of the quadratic residues. For example, computing square roots in $\mathbb{QR}_N^+$ is also equivalent to factoring the modulus

N. We therefore have a "gap group" [29], in which the computational problem (i.e., computing a square root) is as hard as factoring, whereas the corresponding decisional problem (i.e., deciding if an element is a *signed* square) is easy. We can apply this observation to the Diffie-Hellman assumption. Namely, we extend Shmuely's result to show that in the group of signed quadratic residues, the *Strong* Diffie-Hellman problem is implied by the factoring assumption.

Concretely, the signed quadratic residues, $\mathbb{QR}_N^+$, are defined as the group $\mathbb{QR}_N^+ := \{|x| \ : \ x \in \mathbb{QR}_N\}$, where $|x|$ is the absolute value when representing elements of $\mathbb{Z}_N$ as the set $\{-(N-1)/2, \ldots, (N-1)/2\}$. We have that $(\mathbb{QR}_N^+, \circ)$ is a cyclic group, where the group operation is given by $a \circ b := |a \cdot b \bmod N|$. As already noted in [16], membership in $\mathbb{QR}_N^+$ can be efficiently verified since $\mathbb{QR}_N^+ = \mathbb{J}_N^+$, where $\mathbb{J}_N$ is the group of elements with Jacobi symbol 1 and $\mathbb{J}_N^+ := \{|x| \ : \ x \in \mathbb{J}_N\} = \mathbb{J}_N/\pm 1$.

1.3 Hybrid ElGamal over the Signed Quadratic Residues

The Hybrid ElGamal encryption scheme combines the original ElGamal encryption scheme with a hash function for key derivation and a symmetric cipher. As "Diffie-Hellman integrated encryption scheme" (DHIES) [1] it is contained in several standards bodies for public-key encryption, e.g., in IEEE P1363a, SECG, and ISO 18033-2. We consider the security of Hybrid ElGamal when implemented over the group of signed quadratic residues.

CCA security in the random oracle model under the factoring assumption. It is well known [1,12] that Hybrid ElGamal is CCA secure in the random oracle model under the SDH assumption. Recall that we show that the SDH assumption in the group of signed quadratic residues is implied by the factoring assumption. Hence, as an immediate application of our results, we obtain that Hybrid ElGamal over the signed quadratic residues is CCA secure in the random oracle model under the factoring assumption. (We emphasize that while the security proofs for Hybrid ElGamal from [1,12] are formulated for prime-order subgroups of $\mathbb{Z}_p^*$, they do *not* use knowledge about the order of the platform group, and hold literally in the group of signed quadratic residues.)

CCA security in the standard model under the higher residuosity assumption. Using completely different techniques, we show the Hybrid ElGamal over the signed quadratic residues is CCA secure in the standard model under the *higher residuosity* assumption [19].[1] This result is obtained by applying the recent "randomness extraction framework" by [22] to a specific "high-entropic" hash proof system whose subset membership problem is hard assuming the higher residuosity assumption. We stress that this is the first security result for Hybrid ElGamal in the standard model from a non-interactive computational assumption.

[1] The higher residuosity assumption states that it is hard to distinguish random elements of $\mathbb{QR}_N$ from random elements of $\mathbb{G}_S$, where $\mathbb{G}_S$ is a subgroup of $\mathbb{QR}_N$ of unknown (large) order S.

1.4 Other Applications

Security of Diffie-Hellman key exchange. Similar to the Hybrid ElGamal scheme, the (hashed) Diffie-Hellman key exchange protocol [13] can be proven secure against active attacks in the random oracle model under the SDH assumption ([9, Theorem 5]). As with Hybrid ElGamal, the security proof does not use knowledge about the order of the platform group, and hence holds literally over the signed quadratic residues. In particular, we can employ our result about the SDH assumption in the group of signed quadratic residues. We get that the (hashed) Diffie-Hellman key exchange protocol is secure against active attacks in the random oracle model under the factoring assumption, when implemented over the signed quadratic residues.

Simplifying security proofs. As hinted above, encryption schemes that are already formulated over the quadratic residues have to take into account that the set of quadratic residues is not (or, rather, not known to be) efficiently recognizable. In particular, e.g., ciphertexts submitted for decryption may be non-squares. The usual way to deal with this problem is to first square the group elements supplied to decryption, and to "make up for this additional squaring" in the subsequent processing. Additionally, these works already propose to restrict the set of allowed ciphertexts to *signed* quadratic residues (e.g., to prevent an adversary to submit both C and $-C$ for decryption). Hence, the group of signed quadratic residues is implicitly used, but only to "transport" quadratic residues. Our proposal here is to work in the group of signed quadratic residues altogether, whenever a reduction to the factoring assumption is desired. Because the group of signed quadratic residues is efficiently recognizable, this avoids the extra squaring step and the connected complications. In particular, we can simplify both scheme and security proof of the CCA-secure encryption scheme from [21]. This results in a slight efficiency gain, since we save a few modular squarings. We stress that these modifications do not affect the actual reduction to factoring.[2]

1.5 Related Work

To the best of our knowledge, the group of signed quadratic residues appears first in [16] in the context of hard-core bits for generalized Rabin functions. Furthermore, as explained above, it has been used implicitly in several encryption schemes to "transport" quadratic residues, e.g., in [26,8,21]. The security of Hybrid ElGamal has been investigated in [12,25] in the random oracle model, and in [1] in the standard model. In particular, the latter work derives CCA security results for Hybrid ElGamal under the (interactive) "oracle Diffie-Hellman" assumption. The (non-interactive) computational assumption that we employ to show CCA security of Hybrid ElGamal has been suggested and used in [17,6],

[2] It is easy to see that squaring is a one-way permutation (as hard to invert as factoring N) also in the signed quadratic residues. Furthermore, the least significant bit of the squaring function (over the signed quadratic residues) is hard-core, see [16] who consider the "absolute Rabin function E_N^a."

also with the goal to construct hash proof systems for the use in encryption schemes. However, the encryption schemes from [17,6] are less efficient than Hybrid ElGamal due to the fact that they do not use randomness extraction techniques, but instead build on the Cramer-Shoup, resp. Kurosawa-Desmedt paradigms [11,23]. The paper [9] has a similar overall goal as ours. They propose the "Twin Diffie-Hellman" (2DH) assumption and show that the (interactive) *Strong* 2DH assumption is implied by the *standard* DH assumption. However, to be able to use this new assumption to prove security of the schemes of interest (among others also the Hybrid ElGamal and the Diffie-Hellman key-exchange protocol) they have to modify the schemes. Our results directly yield a security proof for the above schemes when instantiated in the specific group of signed quadratic residues.

2 Preliminaries

2.1 Notation

If $k \in \mathbb{N}$ then 1^k denotes the string of k ones. If $r \geq 1$ is a rational number then $[r] = \{1, \ldots, \lceil r \rceil\}$. If S is a set then $s \leftarrow_R S$ denotes the operation of picking an element s of S uniformly at random. We write $\mathcal{A}(x, y, \ldots)$ to indicate that $\mathcal{A}$ is an algorithm with inputs $x, y, \ldots$ and by $z \leftarrow_R \mathcal{A}(x, y, \ldots)$ we denote the operation of running $\mathcal{A}$ with inputs $(x, y, \ldots)$ and letting z be the output. We write $\lg x$ for logarithms over the reals with base 2. The *min-entropy* of a random variable X is defined as $H_\infty(X) = -\lg(\max_{x \in \mathcal{X}} \Pr[X = x])$. If X is an element of a cyclic group $\mathbb{G} = \langle g \rangle$, we write $\mathrm{dlog}_g X$ for the smallest non-negative integer x with $X = g^x$.

2.2 Public-Key Encryption

A *public key encryption* scheme $\mathsf{PKE} = (\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}(k)$ consists of three polynomial time algorithms (PTAs), of which the first two, Kg and Enc, are probabilistic and the last one, Dec, is deterministic. Public/secret keys for security parameter $k \in \mathbb{N}$ are generated using $(pk, sk) \leftarrow_R \mathsf{Kg}(1^k)$. Given such a key pair, a message $m \in \mathcal{M}(k)$ is encrypted by $C \leftarrow_R \mathsf{Enc}(pk, m)$; a ciphertext is decrypted by $m \leftarrow_R \mathsf{Dec}(sk, C)$, where possibly Dec outputs $\perp$ to denote an invalid ciphertext. For consistency, we require that for all $k \in \mathbb{N}$, all messages $m \in \mathcal{M}(k)$, it must hold that $\Pr[\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) = m] = 1$ where the probability is taken over the above randomized algorithms and $(pk, sk) \leftarrow_R \mathsf{Kg}(1^k)$.

The security we require for PKE is $\mathsf{IND\text{-}CCA}$ security [31,14]. We define the advantage of an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ as

$$\mathsf{Adv}^{\mathrm{cca}}_{\mathsf{PKE},\mathcal{A}}(k) \stackrel{\mathrm{def}}{=} \left| \Pr\left[\hat{b} = b \ : \begin{array}{l} (pk, sk) \leftarrow_R \mathsf{Kg}(1^k) \\ (m_0, m_1, St) \leftarrow_R \mathcal{A}_1^{\mathsf{Dec}(sk,\cdot)}(pk) \\ b \leftarrow_R \{0,1\}\,;\ C^* \leftarrow_R \mathsf{Enc}(pk, m_b) \\ b' \leftarrow_R \mathcal{A}_2^{\mathsf{Dec}(sk,\cdot)}(C^*, St) \end{array} \right] - \frac{1}{2} \right| .$$

The adversary $\mathcal{A}_2$ is restricted not to query $\mathsf{Dec}(sk, \cdot)$ with C^*. PKE scheme PKE is said to be indistinguishable against chosen-ciphertext attacks ($\mathsf{IND\text{-}CCA}$ secure in short) if the advantage function $\mathsf{Adv}^{\mathrm{cca}}_{\mathsf{PKE},\mathcal{A}}(k)$ is a negligible function in k for all efficient $\mathcal{A}$.

2.3 Symmetric Encryption

A symmetric encryption scheme $\mathsf{SE} = (\mathsf{E}, \mathsf{D})$ is specified by its encryption algorithm E (encrypting $m \in \mathcal{M}(k)$ with keys $S \in \mathcal{K}_{\mathsf{SE}}(k)$) and decryption algorithm D (returning $m \in \mathcal{M}(k)$ or $\perp$). Here we restrict ourselves to deterministic algorithms E and D.

The most common notion of security for symmetric encryption is that of (one-time) ciphertext indistinguishability ($\mathsf{IND\text{-}OT}$), which requires that all efficient adversaries fail to distinguish between the encryptions of two messages of their choice. Another common security requirement is *ciphertext authenticity*. (One-time) ciphertext integrity ($\mathsf{INT\text{-}OT}$) requires that no efficient adversary can produce a new valid ciphertext under some key when given one encryption of a message of his choice under the same key. A symmetric encryption scheme which satisfies *both* requirements simultaneously is called secure in the sense of authenticated encryption ($\mathsf{AE\text{-}OT}$ secure). Symmetric ciphers secure in the sense of $\mathsf{AE\text{-}OT}$ can be constructed (following the encrypt-then-mac approach [2,12]) from a $\mathsf{IND\text{-}OT}$ secure symmetric encryption scheme and a MAC. Note that $\mathsf{AE\text{-}OT}$ security is a stronger notion than one-time chosen-ciphertext security ($\mathsf{IND\text{-}OTCCA}$) [2,12]. Formal definitions and constructions appear in, e.g., [20].

2.4 Hash Functions

Let $\mathcal{H}$ be a family of hash functions $\mathsf{H} : X \to Y$. With $|\mathcal{H}|$ we denote the number of functions in this family and when sampling from $\mathcal{H}$ we assume a uniform distribution. Let $k > 1$ be an integer, the hash-family $\mathcal{H}$ is k-wise independent if for any sequence of distinct elements $x_1, \ldots, x_k \in X$ the random variables $\mathsf{H}(x_1), \ldots, \mathsf{H}(x_k)$, where $\mathsf{H} \leftarrow_R \mathcal{H}$, are independent and uniformly random.

3 The Group of Signed Quadratic Residues

3.1 Quadratic Residues

An n-bit integer $N = PQ$ is called an RSA modulus if P and Q are two distinct $n/2$-bit odd primes. In what follows, we will assume that N is a Blum integer, i.e., an RSA modulus $N = PQ$ such that P and Q are both congruent 3 modulo 4. The group $\mathbb{Z}_N^*$ consists of all elements of $\mathbb{Z}_N$ that have an inverse modulo N. $\mathbb{Z}_N^*$ has order $\phi(N) = (P-1)(Q-1)$, where $\phi(N)$ is Euler's totient function. By $\mathbb{J}_N$ we denote the subgroup of all elements from $\mathbb{Z}_N^*$ with Jacobi symbol 1. $\mathbb{J}_N$ has index 2 in $\mathbb{Z}_N^*$ and has order $(P-1)(Q-1)/2$. Since N is Blum, $-1 \in \mathbb{J}_N$. By $\mathbb{QR}_N$ we denote the group of quadratic residues modulo N. Note that $\mathbb{QR}_N$ is a subgroup of $\mathbb{J}_N$ with index 2 and has order $(P-1)(Q-1)/4$. We remark that recognizing elements in $\mathbb{QR}_N$ is generally believed to be a hard problem (the quadratic residuosity problem).

3.2 Signed Quadratic Residues

Let N be an integer. For $x \in \mathbb{Z}_N$ we define $|x|$ as the absolute value of x, where x is represented as a signed integer in the set $\{-(N-1)/2, \ldots, (N-1)/2\}$. For a sub-group $\mathbb{G}$ of $\mathbb{Z}_N^*$ we define the "signed group", $\mathbb{G}^+$, as the group

$$\mathbb{G}^+ := \{|x| \ : \ x \in \mathbb{G}\}$$

with the following group operation. Namely, for $g, h \in \mathbb{G}^+$ and an integer x we define

$$g \circ h := |g \cdot h \bmod N|, \qquad g^{\underline{x}} := \underbrace{g \circ g \circ \ldots \circ g}_{x \text{ times}} = |g^x \bmod N| \,. \tag{1}$$

More complicated expressions in the exponents are computed modulo the group order, e.g., $g^{\underline{1/2}} = g^{\underline{2^{-1} \bmod \mathrm{ord}(\mathbb{G}^+)}}$. Note that taking the absolute value is a surjective homomorphism from $\mathbb{G}$ to $\mathbb{G}^+$ with trivial kernel if $-1 \notin \mathbb{G}$, and with kernel $\{-1, 1\}$ if $-1 \in \mathbb{G}$.

Let N be a Blum integer such that $-1 \notin \mathbb{QR}_N$. We will mainly be interested in $\mathbb{QR}_N^+$, which we call *signed quadratic residues* (modulo N). $\mathbb{QR}_N^+$ is a subgroup of $\mathbb{Z}_N^*/\pm 1$, with absolute values as a convenient computational representation. The following basic facts have already been noted in [16].

Lemma 1. *Let N be a Blum integer. Then:*
1. *$(\mathbb{QR}_N^+, \circ)$ is a group of order $\phi(N)/4$.*
2. *$\mathbb{QR}_N^+ = \mathbb{J}_N^+$. In particular, $\mathbb{QR}_N^+$ is efficiently recognizable (given only N).*
3. *If $\mathbb{QR}_N$ is cyclic, so is $\mathbb{QR}_N^+$.*

Proof. First, note that $|\cdot| : (\mathbb{Z}_N, \cdot) \to (\mathbb{Z}_N^+, \circ)$ is a group homomorphism so $(\mathbb{QR}_N^+, \circ)$ is a group. Since $-1 \notin \mathbb{QR}_N$, the map $\mathbb{QR}_N \to \mathbb{QR}_N^+$ has kernel $\{1\}$, and so $\mathrm{ord}(\mathbb{QR}_N^+) = \mathrm{ord}(\mathbb{QR}_N) = \phi(N)/4$. On the other hand, the map $\mathbb{J}_N \to \mathbb{J}_N^+$ has kernel $\{\pm 1\}$, and so $\mathrm{ord}(\mathbb{J}_N^+) = \mathrm{ord}(\mathbb{J}_N)/2 = \phi(N)/4$. Since $\mathbb{QR}_N \subseteq \mathbb{J}_N$, we have $\mathbb{QR}_N^+ \subseteq \mathbb{J}_N^+$, so $\mathrm{ord}(\mathbb{QR}_N^+) = \mathrm{ord}(\mathbb{J}_N^+)$ implies $\mathbb{QR}_N^+ = \mathbb{J}_N^+$. Elements in $\mathbb{QR}_N^+$ can be efficiently recognized since $\mathbb{QR}_N^+ = \mathbb{J}_N^+ = \mathbb{J}_N \cap [(N-1)/2]$. If $\mathbb{QR}_N$ is cyclic, a generator g of $\mathbb{QR}_N$ is mapped to a generator $|g|$ of $\mathbb{QR}_N^+$, so $\mathbb{QR}_N^+$ is a cyclic group.

3.3 Factoring Assumption

RSA Instance Generator. Let $0 \le \delta < 1/2$ be a constant and $n(k)$ be a function. Let RSAgen be an algorithm that generates elements (N, P, Q), such that $N = PQ$ is an n-bit Blum integer and all prime factors of $\phi(N)/4$ are pairwise distinct and at least δn bit integers.[3]

[3] The "only large prime-factors" requirement is needed to ensure that the square of a random element in $\mathbb{Z}_N^*$ is a generator of $\mathbb{QR}_N$ with high probability $1 - O(2^{-\delta n(k)})$. The requirement that all prime factors are distinct ensures that $\mathbb{J}_N$ is cyclic.

FACTORING ASSUMPTION. The *factoring assumption* is that computing P, Q from N (generated by RSAgen) is hard. We write

$$\mathsf{Adv}^{\mathrm{fac}}_{\mathcal{A},\mathsf{RSAgen}}(k) := \Pr[\{P,Q\} \leftarrow_R \mathcal{A}(N) \ : \ (N,P,Q) \leftarrow_R \mathsf{RSAgen}(1^k)].$$

The factoring assumption for RSAgen holds if $\mathsf{Adv}^{\mathrm{fac}}_{\mathcal{A},\mathsf{RSAgen}}(k)$ is negligible for all efficient $\mathcal{A}$.

3.4 Strong Diffie-Hellman Assumption

Let $\mathbb{G}$ be a finite cyclic group whose order is not necessarily known. The Diffie-Hellman (DH) problem in $\mathbb{G}$ is to compute $\mathrm{DH}_g(X,Y) := g^{(\mathrm{dlog}_g X)(\mathrm{dlog}_g Y)}$ from $(\mathbb{G}, g, X, Y)$ for a uniform generator g and uniform $X, Y \in \mathbb{G}$. The strong Diffie-Hellman problem [1] is the same as the DH problem, but now the adversary has access to a Decision Diffie-Hellman oracle for fixed g and X, which is defined as $\mathrm{DDH}_{g,X}(\hat{Y},\hat{Z}) = 1$ if $\hat{Y}^{\mathrm{dlog}_g X} = \hat{Z}$ (and $\mathrm{DDH}_{g,X}(\hat{Y},\hat{Z}) = 0$ else), where $(\hat{Y},\hat{Z}) \in \mathbb{G} \times \mathbb{G}$. We do not define $\mathrm{DDH}_{g,X}$ in inputs $(\hat{Y},\hat{Z}) \notin \mathbb{G} \times \mathbb{G}$, since we assume that $\mathbb{G}$ is efficiently recognizable. For our purposes, we will consider the group $(\mathbb{QR}^+_N, \circ)$, i.e., the group of signed quadratic residues.

To an adversary $\mathcal{A}$ and RSAgen we associate

$$\mathsf{Adv}^{\mathrm{sdh}}_{\mathcal{A},\mathsf{RSAgen}}(k) := \Pr\left[Z = \mathrm{DH}_g(X,Y) \ : \ \begin{array}{l} (N,P,Q,S) \leftarrow_R \mathsf{RSAgen}(1^k)\,; \\ \text{unif. choose } g \text{ with } \langle g \rangle = \mathbb{QR}^+_N\,; \\ X, Y \leftarrow_R \mathbb{QR}^+_N\,; \\ Z \leftarrow_R \mathcal{A}^{\mathrm{DDH}_{g,X}(\cdot,\cdot)}(N,g,X,Y) \end{array} \right].$$

The Strong DH assumption holds relative to RSAgen if $\mathsf{Adv}^{\mathrm{sdh}}_{\mathcal{A},\mathsf{RSAgen}}(k)$ is negligible for all efficient $\mathcal{A}$.

Theorem 2. *If the factoring assumption holds then the strong DH assumption holds relative to* RSAgen*. In particular, for every strong DH adversary* $\mathcal{A}$*, there exists a factoring adversary* $\mathcal{B}$ *(with roughly the same complexity as* $\mathcal{A}$*) such that*

$$\mathsf{Adv}^{\mathrm{sdh}}_{\mathcal{A},\mathsf{RSAgen}}(k) \le \mathsf{Adv}^{\mathrm{fac}}_{\mathcal{B},\mathsf{RSAgen}}(k) + O(2^{-\delta n(k)}). \tag{2}$$

Proof. We construct $\mathcal{B}$ from a given $\mathcal{A}$. Concretely, $\mathcal{B}$ receives a challenge $N = PQ$, chooses uniformly $u \leftarrow_R (\mathbb{Z}^*_N)^+ \setminus \mathbb{QR}^+_N$ and sets $h := u^{\underline{2}}$. Note that by definition of N, we have $\langle h \rangle = \mathbb{QR}^+_N$ except with probability $O(2^{-\delta n(k)})$. Then $\mathcal{B}$ chooses $a, b \in [N/4]$ and sets

$$g := h^{\underline{2}} \qquad X := h \circ g^{\underline{a}} \qquad Y := h \circ g^{\underline{b}}.$$

This implicitly defines

$$\mathrm{dlog}_g X = a + 1/2 \bmod \mathrm{ord}(\mathbb{QR}^+_N), \quad \text{and} \quad \mathrm{dlog}_g Y = b + 1/2 \bmod \mathrm{ord}(\mathbb{QR}^+_N),$$

where the discrete logarithms are of course considered in $(\mathbb{QR}^+_N, \circ)$. Again, by definition of N, the statistical distance between these (g, X, Y) and the input of

$\mathcal{A}$ in the strong DH experiment is bounded by $O(2^{-\delta n(k)})$. So $\mathcal{B}$ runs $\mathcal{A}$ on input (g, X, Y), and answers $\mathcal{A}$'s oracle queries $(\hat{Y}, \hat{Z})$ as follows. First, we may assume that $\hat{Y}, \hat{Z} \in \mathbb{QR}_N^+$ since (by Lemma 1) $\mathbb{QR}_N^+ = \mathbb{J}_N^+$ is efficiently recognizable. Next, since N is a Blum integer, the group order $\mathrm{ord}(\mathbb{QR}_N^+) = (P-1)(Q-1)/4$ is odd, and hence

$$\hat{Y}^{\underline{\mathrm{dlog}_g X}} = \hat{Z} \quad \Longleftrightarrow \quad \hat{Y}^{\underline{2\mathrm{dlog}_g X}} = \hat{Z}^{\underline{2}} \quad \Longleftrightarrow \quad \hat{Y}^{\underline{2a+1}} = \hat{Z}^{\underline{2}}.$$

Thus, $\mathcal{B}$ can implement the strong DH oracle by checking whether $\hat{Y}^{\underline{2a+1}} \stackrel{?}{=} \hat{Z}^{\underline{2}}$.

Consequently, with probability $\mathsf{Adv}^{\mathrm{sdh}}_{\mathcal{A},\mathsf{RSAgen}}(k) - O(2^{-\delta n(k)})$, $\mathcal{A}$ will finally output

$$Z = g^{\underline{(\mathrm{dlog}_g X)(\mathrm{dlog}_g Y)}} = g^{\underline{(a+1/2)(b+1/2)}} = h^{\underline{2ab+a+b+1/2}} \in \mathbb{QR}_N^+,$$

from which $\mathcal{B}$ can extract $v := h^{\underline{1/2}} \in \mathbb{QR}_N^+$ (using its knowledge about a and b). Since $u \notin \mathbb{QR}_N^+$ and $v \in \mathbb{QR}_N^+$ are two non-trivially different square roots of h, $\mathcal{B}$ can factor N by computing $\gcd(u - v, N)$.

4 Hybrid ElGamal over the Signed Quadratic Residues

We recall the Hybrid ElGamal (aka DHIES) scheme from [1,12]. There the scheme is described in a more general form over arbitrary cyclic groups. Here we restrict ourselves to the special case of $\mathbb{QR}_N^+$, for the following choice of N:

RSA Instance Generator. Let $0 \leq \delta \leq 1/4$ be a constant and $n(k)$ be a function. Let $\mathsf{RSAgen}' = \mathsf{RSAgen}'_{\delta,n(k)}$ be an algorithm that generates elements (N, P, Q, S), such that

- $N = PQ$ is an n-bit Blum integer such that the prime factors of $\phi(N)/4$ are pairwise distinct and at least δn-bit integers;
- $S > 1$ is a divisor of $\phi(N)/4$ with $1 < \gcd(S, (P-1)/2) < (P-1)/2$ and $1 < \gcd(S, (Q-1)/2) < (Q-1)/2$ (so S splits up into large prime factors of both $(P-1)/2$ and $(Q-1)/2$, but such that neither $(P-1)/2$ nor $(Q-1)/2$ divides S).

Note that by construction, $\gcd(S, \phi(N)/(4S)) = 1$. We stress that we need this choice of N *only* for the security proof of Hybrid ElGamal in the standard model. The security proof in the random oracle model (based on the hardness of factoring N) works with RSA instances as generated by RSAgen' or RSAgen.

4.1 The Encryption Scheme

Let $\mathsf{SE} = (\mathsf{E}, \mathsf{D})$ be a symmetric cipher with key-space $\{0,1\}^{\ell(k)}$, let $\mathcal{H} = (\mathcal{H}_k)_{k\in\mathbb{N}}$ be a family of hash functions with $\mathsf{H} : \{0,1\}^{2n(k)} \to \{0,1\}^{\ell(k)}$ for each $\mathsf{H} \in \mathcal{H}_k$. Define the following encryption scheme $\mathsf{DHIES} = (\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$:

Key generation. $\mathsf{Kg}(1^k)$ chooses uniformly at random

- an RSA modulus $N = PQ$ generated with $\mathsf{RSAgen}'(1^k)$,
- a generator g of $\mathbb{QR}_N^+$,
- an exponent $x \in [N/4]$,
- a hash function $\mathsf{H} \in \mathcal{H}_k$.

Kg then sets $X = g^{\underline{x}} \in \mathbb{QR}_N^+$ and outputs a public key pk and a secret key sk, where

$$pk = (N, g, X, \mathsf{H}) \qquad\qquad sk = (N, x, \mathsf{H}).$$

Encryption. $\mathsf{Enc}(pk, m)$ chooses uniformly $y \in [N/4]$, sets

$$Y = g^{\underline{y}} \qquad\qquad K = \mathsf{H}(Y, X^{\underline{y}}) \qquad\qquad \psi = \mathsf{E}_K(m)$$

and outputs the ciphertext $(Y, \psi) \in \mathbb{QR}_N^+ \times \{0,1\}^*$.

Decryption. $\mathsf{Dec}(sk, (Y, \psi))$ verifies that $Y \in \mathbb{QR}_N^+$ and rejects if not. Then, Dec computes $K = \mathsf{H}(Y, Y^{\underline{x}})$ and outputs $\mathsf{D}_K(\psi)$.

Note that we present the DHIES scheme in a slightly generalized form for general symmetric ciphers SE, whereas in [1], SE consisted of a particular "encrypt-then-mac"-based cipher (which is AE-OT and therefore also IND-OTCCA secure).

4.2 Security

We now state our claims about the security of DHIES. We will prove that the same scheme DHIES is secure in the standard and in the random oracle model, under different assumptions.

Theorem 3. *Assume the factoring assumption holds for* $\mathsf{RSAgen}'_{n(k),\delta}$*,* $\mathcal{H}$ *is modeled as a random oracle, and* SE *is* IND-OTCCA *secure. Then* DHIES *is* IND-CCA *secure.*

[12, Theorem 9] show that the IND-CCA security of hashed ElGamal (viewed as a key encapsulation mechanism) in the random oracle model is implied by the strong DH assumption. In Theorem 9 (Appendix A) we formally show that their result does not use a specific group structure and can also be applied to our case. Putting Theorem 2 and Theorem 9 together yields Theorem 3. The following theorem will be proved in Section 5.

Theorem 4. *Assume the Higher Residuosity assumption (to be introduced in Section 5) holds relative to* $\mathsf{RSAgen}'_{\delta,n(k)}$*,* $\mathcal{H}$ *is a family of 4-wise independent hash functions, and* SE *is* AE-OT *secure with* ℓ*-bit keys. If* $\delta n(k) \geq 4\ell$*, then* DHIES *is* IND-CCA *secure.*

5 A Security Proof in the Standard Model

5.1 The Computational Hardness Assumption

To prove the security of DHIES in the standard model, we will make use of the following hardness assumption.

Let (N, P, Q, S) be generated by RSAgen'. We write $\mathbb{G}_S$ for the unique subgroup of order S of $\mathbb{Z}_N^*$. The higher residuosity (HR) assumption states that distinguishing a random element from $\mathbb{G}_S$ from a random element from $\mathbb{QR}_N$ is computationally infeasible. More formally, to an adversary and RSAgen' we associate

$$\mathsf{Adv}^{\mathrm{hr}}_{\mathcal{A},\mathsf{RSAgen}'}(k) := |\Pr[1 \leftarrow_R \mathcal{A}(N, g, c)] - \Pr[1 \leftarrow_R \mathcal{A}(N, g, \tilde{c})]|\,,$$

where $(N, P, Q, S) \leftarrow_R \mathsf{RSAgen}'(1^k)$, $g, c \leftarrow_R \mathbb{G}_S$ and $\tilde{c} \leftarrow_R \mathbb{QR}_N$. The HR assumption for RSAgen' holds if $\mathsf{Adv}^{\mathrm{hr}}_{\mathcal{A},\mathsf{RSAgen}'}(k)$ is negligible for all efficient $\mathcal{A}$. Note that the HR assumption implicitly depends on the choice of $n(k)$ and δ. For concreteness, for $k = 80$ bits security one may choose $n(k) = 1024$ and $\delta = 1/8$. Then N can be sampled as $N = PQ$ for $P = 2P_SP_T + 1$ and $Q = 2Q_SQ_T + 1$ for primes P_S, P_T, Q_S, Q_T, with $P_S, Q_T \approx 2^{\delta n}$, such that for $S = P_SQ_S$, the order of $\mathbb{G}_S$ is about 2^{256}.

In the literature several related assumptions can be found. Closest to our assumption are the ones in [19,24,17,6] which are as our HR assumption but with a different distribution of N and/or using the groups $\mathbb{J}_N, \mathbb{Z}_N^*$ instead of $\mathbb{QR}_N$. Other similar assumptions were proposed in [18,10,24,4,28]. In all these assumptions the adversary is given (N, S) where $S \mid \phi(N)/4$, and has to distinguish a "random element" from one of the form $x^S \bmod N$.

5.2 A Variant of DHIES

To prove Theorem 4, we will consider a slightly different scheme, $\mathsf{DHIES}' = (\mathsf{Kg}', \mathsf{Enc}, \mathsf{Dec})$. It is defined as DHIES, with the only difference that in Kg', the element g from key generation is a uniform element from $\mathbb{G}_S^+$ (instead of an uniform element from $\mathbb{QR}_N^+$). The following lemma is immediate.

Lemma 5. *Under the HR assumption,* DHIES *is* $\mathsf{IND\text{-}CCA}$ *if and only if* DHIES' *is* $\mathsf{IND\text{-}CCA}$. *In particular, for every adversary* $\mathcal{A}$ *there exists an adversary* $\mathcal{B}$ *with*

$$|\mathsf{Adv}^{\mathrm{cca}}_{\mathsf{DHIES},\mathcal{A}}(k) - \mathsf{Adv}^{\mathrm{cca}}_{\mathsf{DHIES}',\mathcal{A}}(k)| \leq \mathsf{Adv}^{\mathrm{hr}}_{\mathsf{RSAgen}',\mathcal{B}}(k).$$

Lemma 6. *Under the conditions from Theorem 4,* DHIES' *is* $\mathsf{IND\text{-}CCA}$ *secure.*

A combination of the above two lemmas yields Theorem 4. The rest of this section is devoted to the proof of Lemma 6.

5.3 Hash Proof Systems

We recall the notion of hash proof systems introduced by Cramer and Shoup [11].

Smooth Projective Hashing. Let $\mathcal{C}, \mathcal{K}$ be sets and $\mathcal{V} \subset \mathcal{C}$ a language. In the context of public-key encryption (and viewing a hash proof system as a key-encapsulation mechanism (KEM) [12] with "special algebraic properties") one may think of $\mathcal{C}$ as the set of all *ciphertexts*, $\mathcal{V} \subset \mathcal{C}$ as the set of all *valid (consistent) ciphertexts*, and $\mathcal{K}$ as the set of all *symmetric keys*. Let $\Lambda_{sk} : \mathcal{C} \to \mathcal{K}$

be a hash function indexed with $sk \in \mathcal{SK}$, where $\mathcal{SK}$ is a set. A hash function Λ_{sk} is *projective* if there exists a projection $\mu : \mathcal{SK} \to \mathcal{PK}$ such that $\mu(sk) \in \mathcal{PK}$ defines the action of Λ_{sk} over the subset $\mathcal{V}$. That is, for every $C \in \mathcal{V}$, the value $K = \Lambda_{sk}(C)$ is uniquely determined by $\mu(sk)$ and C. In contrast, nothing is guaranteed for $C \in \mathcal{C} \setminus \mathcal{V}$, and it may not be possible to compute $\Lambda_{sk}(C)$ from $\mu(sk)$ and C. Following [22] we make the following two definitions about projective hash functions. The projective hash function is κ-*entropic* if for all $C \in \mathcal{C} \setminus \mathcal{V}$, $H_\infty(\Lambda_{sk}(C) \mid pk) \geq \kappa$ where in the above $pk = \mu(sk)$ for $sk \leftarrow_R \mathcal{SK}$. We furthermore define the collision probability as $\delta = \max_{C, C^* \in \mathcal{C} \setminus \mathcal{V}, C \neq C^*}(\Pr_{sk}[\Lambda_{sk}(C) = \Lambda_{sk}(C^*)])$.

Hash Proof System. A hash proof system $\mathsf{HPS} = (\mathsf{Par}, \mathsf{Pub}, \mathsf{Priv})$ consists of three algorithms. The randomized algorithm $\mathsf{Par}(1^k)$ generates parametrized instances of $par = (group, \mathcal{K}, \mathcal{C}, \mathcal{V}, \mathcal{PK}, \mathcal{SK}, \Lambda_{(\cdot)} : \mathcal{C} \to \mathcal{K}, \mu : \mathcal{SK} \to \mathcal{PK})$, where *group* may contain some additional structural parameters. The deterministic public evaluation algorithm Pub inputs the projection key $pk = \mu(sk)$, $C \in \mathcal{V}$ and a witness r of the fact that $C \in \mathcal{V}$ and returns $K = \Lambda_{sk}(C)$. The deterministic private evaluation algorithm Priv inputs $sk \in \mathcal{SK}$ and returns $\Lambda_{sk}(C)$, without knowing a witness. We further assume that μ is efficiently computable and that there are efficient algorithms given for sampling $sk \in \mathcal{SK}$, sampling $C \in \mathcal{V}$ uniformly (or negligibly close to) together with a witness r, sampling $C \in \mathcal{C}$ uniformly (given sk), and for checking membership in $\mathcal{C}$. Following [23] we also require that the subset membership problem can be efficiently solved with a master trapdoor.

Subset Membership Problem. As computational problem we require that the *subset membership problem* is hard in HPS. That is, for random $C_0 \in \mathcal{V}$ and random $C_1 \in \mathcal{C} \setminus \mathcal{V}$ the two elements C_0 and C_1 are computationally indistinguishable. This is captured by defining the advantage function $\mathsf{Adv}^{\mathrm{sm}}_{\mathsf{HPS},\mathcal{A}}(k)$ of an adversary $\mathcal{A}$ as

$$\mathsf{Adv}^{\mathrm{sm}}_{\mathsf{HPS},\mathcal{A}}(k) \stackrel{\mathrm{def}}{=} \left|\Pr[1 \leftarrow_R \mathcal{A}(\mathcal{C}, \mathcal{V}, C_1)] - \Pr[1 \leftarrow_R \mathcal{A}(\mathcal{C}, \mathcal{V}, C_0)]\right|$$

where $\mathcal{C}$ is taken from the output of $\mathsf{Par}(1^k)$, $C_1 \leftarrow_R \mathcal{C}$ and $C_0 \leftarrow_R \mathcal{C} \setminus \mathcal{V}$.

5.4 IND-CCA Secure Encryption via Randomness Extraction

We recall the randomness extraction framework [22] that (building on [23,11]) transforms any κ-entropic HPS with hard subset membership problem into a IND-CCA secure encryption scheme.

Let $\mathsf{HPS} = (\mathsf{Par}, \mathsf{Pub}, \mathsf{Priv})$ be a hash proof system, let $\mathcal{H}$ be a family of hash functions with $\mathsf{H} : \mathcal{K} \to \{0,1\}^{\ell(k)}$ and let $\mathsf{SE} = (\mathsf{E}, \mathsf{D})$ be an AE-OT secure symmetric encryption scheme with key-space $\mathcal{K}_{\mathsf{SE}} = \{0,1\}^{\ell(k)}$. We build a public-key encryption scheme $\mathsf{PKE}_{\mathsf{HPS}} = (\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ as follows.

Key generation. $\mathsf{Kg}(1^k)$ picks $par \leftarrow_R \mathsf{Par}(1^k)$, $sk \leftarrow_R \mathcal{SK}$ and defines $pk = \mu(sk) \in \mathcal{PK}$. Next, it picks a random hash function $\mathsf{H} \leftarrow_R \mathcal{H}$. The public-key is (par, H, pk), the secret-key is (par, H, sk).

Encryption. $\mathsf{Enc}(pk, m)$ picks $C \leftarrow_R \mathcal{V}$ together with its witness r that $C \in \mathcal{V}$. Session key $K = \mathsf{H}(\Lambda_{sk}(C)) \in \{0,1\}^\ell$ is computed as $K \leftarrow \mathsf{H}(\mathsf{Pub}(pk, C, r))$. The symmetric ciphertext is $\psi \leftarrow \mathsf{E}_K(m)$. The ciphertext is (C, ψ).

Decryption. $\mathsf{Dec}(sk, C)$ first checks if $C \in \mathcal{C}$ and rejects if not. Otherwise, it reconstructs the session key $K = \mathsf{H}(\Lambda_{sk}(C))$ as $K \leftarrow \mathsf{H}(\mathsf{Priv}(sk, C))$ and returns $\{m, \perp\} \leftarrow \mathsf{D}_K(\psi)$.

Theorem 7. *[22] Assume* HPS *is $\kappa(k)$-entropic with hard subset membership problem and negligible collision probability, $\mathcal{H}$ is a family of 4-wise independent hash functions with $\mathsf{H} : \mathcal{K} \to \{0,1\}^{\ell(k)}$, and* SE *is* AE-OT *secure. If $\kappa(k) \geq 2(\ell(k) + k)$ then $\mathsf{PKE}_{\mathsf{HPS}}$ is secure in the sense of* IND-CCA.

5.5 A Hash Proof System for DHIES′

We now give a hash proof system HPS that yields the encryption scheme DHIES′ via the transformation given in the last subsection. Define *group* $= (N, g)$, where $(N, P, Q, S) \leftarrow_R \mathsf{RSAgen}'(1^k)$ and g is a uniform generator of $\mathbb{G}_S^+$. Recall that N is of bit-length $n(k)$ and S is of bit-length $\delta n(k)$. Define $\mathcal{C} = \mathbb{QR}_N^+$ and $\mathcal{V} = \mathbb{G}_S^+ = \{g^{\underline{r}} \ : \ r \in \mathbb{Z}_S\}$. A value $r \in \mathbb{Z}$ is a witness of $C \in \mathcal{V}$. Note that it is possible to sample an almost uniform element from $\mathcal{V}$ together with a witness by first picking $r \in \mathbb{Z}_{[N/4]}$ and defining $C = g^{\underline{r}} \in \mathbb{G}_S^+$. Furthermore, membership in $\mathcal{C}$ can be efficiently checked by Lemma 1. Define $\mathcal{SK} = [N/4]$, $\mathcal{PK} = \mathbb{G}_S^+$, and $\mathcal{K} = \mathbb{QR}_N^+ \times \mathbb{QR}_N^+$ (which we interpret as a subset of $\{0,1\}^{2n(k)}$). For $sk = x \in [N/4]$, define $\mu(sk) = X = g^{\underline{x}} \in \mathbb{G}_S^+$. This defines the output of $\mathsf{Par}(1^k)$. For $C \in \mathcal{C}$ define

$$\Lambda_{sk}(C) := (C, C^{\underline{x}}) \ .$$

This defines $\mathsf{Priv}(sk, C)$. Given $pk = \mu(sk)$, $C \in \mathcal{V}$ and a witness $r \in \mathbb{Z}$ such that $C = g^{\underline{r}}$, public evaluation $\mathsf{Pub}(pk, C, r)$ computes $K = \Lambda_{sk}(C)$ as

$$K = (g^{\underline{r}}, X^{\underline{r}}) \ .$$

The trapdoor ω is the order of the group $\mathbb{G}_S$. This completes the description of HPS. Note that $\mathsf{PKE}_{\mathsf{HPS}}$ is exactly DHIES′. Therefore the proof Lemma 6 follows by combining Theorem 7 with the following.

Lemma 8. *Under the HR assumption, the subset membership problem is hard in* HPS. *Furthermore,* HPS *is $\delta n(k)$-entropic with collision probability $\delta = 0$.*

Proof. The subset membership problem is hard in HPS by definition of the HR assumption. The collision probability δ is zero since $\Lambda_{sk}(C) = (C, C^{\underline{x}})$ contains the element C. To show that HPS is $\delta n(k)$-entropic we consider an element $C \in \mathcal{C} \setminus \mathcal{V} = \mathbb{QR}_N^+ \setminus \mathbb{G}_S^+$. We can decompose $\mathbb{QR}_N^+$ as an internal direct product $\mathbb{QR}_N^+ = \mathbb{G}_T^+ \times \mathbb{G}_S^+$, where $\mathbb{G}_T^+$ is a cyclic group of order $T = (P-1)(Q-1)/(4S)$ with $\gcd(T, S) = 1$. Since T has only prime factors greater than $2^{\delta n(k)}$, and $C \notin$

$\mathbb{G}_S^+$, we have $\gcd(\mathrm{ord}(C), T) \geq 2^{\delta n(k)}$. Then, given N, g, $pk = \mu(sk) = X = g^{\underline{x}}$, and any $C \in \mathcal{C} \setminus \mathcal{V}$,

$$\begin{aligned} H_\infty((C, C^{\underline{x}}) \mid N, g, pk, C) &= H_\infty(C^{\underline{x}} \mid N, g, g^{\underline{x}}, C) \\ &= H_\infty(x \bmod \mathrm{ord}(C) \mid x \bmod S, S, T) \\ &\geq H_\infty(x \bmod \gcd(\mathrm{ord}(C), T) \mid x \bmod S, S, T) \\ &\overset{\gcd(S,T)=1}{=} H_\infty(x \bmod \gcd(\mathrm{ord}(C), T) \mid T) \geq \delta n(k) \ . \end{aligned}$$

This completes the proof.

5.6 Extensions

If one only requires a scheme that is IND-CCA secure in the standard model from the HR assumption, the one can turn encryption in DHIES′ slightly more efficient by choosing $y \leftarrow_R [2^{\delta n(k)+k}]$ (instead of $y \leftarrow_R [N/4]$). Furthermore, it is possible to prove the DHIES instantiated with RSAgen (instead of RSAgen′) IND-CCA secure under the ϕ-Hiding assumption [7] which essentially says that the two distributions (N, g) and (N', g') are computationally indistinguishable, where $(N, P, Q) \leftarrow_R \mathsf{RSAgen}$, $g \leftarrow_R \mathbb{QR}_N^+$ and $(N', P', Q', S') \leftarrow_R \mathsf{RSAgen}'$, $g' \leftarrow_R \mathbb{G}_{S'}^+$.

Acknowledgements. We thank Victor Shoup and the anonymous reviewers for useful comments.

References

1. Abdalla, M., Bellare, M., Rogaway, P.: The oracle Diffie-Hellman assumptions and an analysis of DHIES. In: Naccache, D. (ed.) CT-RSA 2001. LNCS, vol. 2020, pp. 143–158. Springer, Heidelberg (2001)
2. Bellare, M., Namprempre, C.: Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 531–545. Springer, Heidelberg (2000)
3. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Ashby, V. (ed.) ACM CCS 1993. ACM Press, New York (1993)
4. Benaloh, J.C.: Dense probabilistic encryption. In: SAC 1994, pp. 120–128 (1994)
5. Blum, M., Goldwasser, S.: An efficient probabilistic public-key encryption scheme which hides all partial information. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 289–299. Springer, Heidelberg (1985)
6. Brown, J., Nieto, J.M.G., Boyd, C.: Concrete chosen-ciphertext secure encryption from subgroup membership problems. In: Pointcheval, D., Mu, Y., Chen, K. (eds.) CANS 2006. LNCS, vol. 4301, pp. 1–18. Springer, Heidelberg (2006)
7. Cachin, C., Micali, S., Stadler, M.: Computationally private information retrieval with polylogarithmic communication. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 402–414. Springer, Heidelberg (1999)

8. Camenisch, J., Shoup, V.: Practical verifiable encryption and decryption of discrete logarithms. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 126–144. Springer, Heidelberg (2003)
9. Cash, D., Kiltz, E., Shoup, V.: The Twin Diffie-Hellman problem and applications. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 127–145. Springer, Heidelberg (2008)
10. Cohen, J.D., Fischer, M.J.: A robust and verifiable cryptographically secure election scheme (extended abstract). In: FOCS, pp. 372–382 (1985)
11. 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)
12. Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM Journal on Computing 33(1), 167–226 (2003)
13. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Transactions on Information Theory 22(6), 644–654 (1976)
14. Dolev, D., Dwork, C., Naor, M.: Nonmalleable cryptography. SIAM Journal on Computing 30(2), 391–437 (2000)
15. El Gamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 10–18. Springer, Heidelberg (1985)
16. Fischlin, R., Schnorr, C.-P.: Stronger security proofs for RSA and Rabin bits. Journal of Cryptology 13(2), 221–244 (2000)
17. Gjøsteen, K.: Symmetric subgroup membership problems. In: Vaudenay, S. (ed.) PKC 2005. LNCS, vol. 3386, pp. 104–119. Springer, Heidelberg (2005)
18. Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences 28(2), 270–299 (1984)
19. Groth, J.: Cryptography in subgroups of zn. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 50–65. Springer, Heidelberg (2005)
20. Hofheinz, D., Kiltz, E.: Secure hybrid encryption from weakened key encapsulation. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 553–571. Springer, Heidelberg (2007)
21. Hofheinz, D., Kiltz, E.: Practical chosen ciphertext secure encryption from factoring. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, pp. 313–332. Springer, Heidelberg (2009)
22. Kiltz, E., Pietrzak, K., Stam, M., Yung, M.: A new randomness extraction paradigm for hybrid encryption. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, pp. 589–608. Springer, Heidelberg (2009)
23. Kurosawa, K., Desmedt, Y.: A new paradigm of hybrid encryption scheme. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 426–442. Springer, Heidelberg (2004)
24. Kurosawa, K., Katayama, Y., Ogata, W., Tsujii, S.: General public key residue cryptosystems and mental poker protocols. In: Damgård, I. (ed.) EUROCRYPT 1990. LNCS, vol. 473, pp. 374–388. Springer, Heidelberg (1991)
25. Kurosawa, K., Matsuo, T.: How to remove MAC from DHIES. In: Wang, H., Pieprzyk, J., Varadharajan, V. (eds.) ACISP 2004. LNCS, vol. 3108, pp. 236–247. Springer, Heidelberg (2004)
26. Lucks, S.: A variant of the cramer-shoup cryptosystem for groups of unknown order. In: Zheng, Y. (ed.) ASIACRYPT 2002, vol. 2501, pp. 27–45. Springer, Heidelberg (2002)

27. McCurley, K.S.: A key distribution system equivalent to factoring. Journal of Cryptology 1(2), 95–105 (1988)
28. Naccache, D., Stern, J.: A new public key cryptosystem based on higher residues. In: ACM CCS 1998, pp. 59–66. ACM Press, New York (1998)
29. Okamoto, T., Pointcheval, D.: The gap-problems: A new class of problems for the security of cryptographic schemes. In: Kim, K. (ed.) PKC 2001. LNCS, vol. 1992, pp. 104–118. Springer, Heidelberg (2001)
30. Rabin, M.O.: Digital signatures and public key functions as intractable as factorization. Technical Report MIT/LCS/TR-212, Massachusetts Institute of Technology (January 1979)
31. Rackoff, C., Simon, D.R.: Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 433–444. Springer, Heidelberg (1992)
32. Shmuely, Z.: Composite diffie-hellman public-key generating systems are hard to break. Technical Report 356, Computer Science Department, Technion, Israel (1985)

A A Security Proof in the Random Oracle Model

Theorem 9. *([12,1]) If the strong DH assumption holds relative to* RSAgen'*, and if* SE *is an* IND-CCA *secure symmetric cipher, then* DHIES *is* IND-IND-OTCCA *secure in the random oracle model. In particular, for every adversary* $\mathcal{A}$ *on* DHIES*, there exist adversaries* $\mathcal{B}$*, resp.* $\mathcal{B}'$ *on the strong DH assumption, resp. the* IND-IND-OTCCA *security of* SE*, such that* $\mathcal{B}$ *and* $\mathcal{B}'$ *have roughly the same complexity as* $\mathcal{A}$*, and*

$$\mathsf{Adv}^{\mathrm{cca}}_{\mathcal{A},\mathsf{DHIES}}(k) \leq \mathsf{Adv}^{\mathrm{sdh}}_{\mathcal{B},\mathsf{RSAgen}'}(k) + \mathsf{Adv}^{\mathrm{cca}}_{\mathcal{B}',\mathsf{SE}}(k) + O(2^{-\delta n(k)}).$$

The adaptations to [12, Theorem 9] are merely syntactic, and below we provide a short proof sketch. Putting Theorem 2 and Theorem 9 together yields Theorem 3.

Proof (Theorem 9). (Sketch.) We proceed in games.

Game 0. Let Game 0 be the original IND-CCA experiment with scheme DHIES and adversary $\mathcal{A}$. Here and in the following games, p_i denotes the probability that the experiment outputs 1, i.e., that $b = \hat{b}$, in Game i. By definition,

$$\mathsf{Adv}^{\mathrm{cca}}_{\mathcal{A},\mathsf{PKE}} = |p_0 - 1/2|. \tag{3}$$

Game 1. In Game 1, we modify the encryption of the challenge ciphertext (Y^*, ψ^*). Namely, now the symmetric ciphertext ψ^* is generated with an independent, uniform symmetric key K' as $\psi^* := \mathsf{E}_{K'}(m_b)$. Decryption queries of the form (Y^*, ψ) (for arbitrary $\psi \neq \psi^*$) are treated as if Y^* decrypted to key K' (and not key $K^* = \mathsf{H}(Y^*, Z^*)$ for $Z^* = Y^{*\underline{r}}$). Let F denote the event that $\mathcal{A}$ queries the random oracle H with (Y^*, Z^*). (F is defined in both Game 0 and Game 1.) Note that the views of $\mathcal{A}$ are *identical* in Game 0 and Game 1 unless F occurs. Hence,

$$|p_1 - p_0| \leq \Pr[F]. \tag{4}$$

Now we can build an adversary $\mathcal{B}$ on the strong DH assumption with

$$\Pr[F] \leq \mathsf{Adv}^{\mathrm{sdh}}_{\mathcal{B},\mathsf{RSAgen}'}(k) + O(2^{-\delta n(k)}). \tag{5}$$

Concretely, $\mathcal{B}^{\mathrm{DDH}_{g,X}(\cdot,\cdot)}(N, g, X, Y^*)$ simulates Game 1 with public key $pk := (N, g, X, \mathsf{H})$, and challenge ciphertext $(Y^*, \psi^*) := (Y, \mathsf{E}_{K'}(m_b))$. Adversary $\mathcal{A}$'s decryption queries $(\hat{Y}, \hat{\psi})$ are answered as follows (note that $\mathcal{B}$ does not know the secret key $x = \mathrm{dlog}_g X$, and hence cannot decrypt directly). If $\mathcal{A}$ has already made an H-query $\mathsf{H}(\hat{Y}, \hat{Z})$ for which $\mathrm{DDH}_{g,X}(\hat{Y}, \hat{Z}) = 1$, then $\hat{Z} = \hat{Y}^{\underline{x}}$, so the key $\hat{K} := \mathsf{H}(\hat{Y}, \hat{Z})$ can be used to decrypt $\hat{\psi}$. If on the other hand $\mathcal{A}$ made no such query, the hash value $\mathsf{H}(\hat{Y}, \hat{Z})$ for the "right" $\hat{Z} = \hat{Y}^{\underline{x}}$ has not yet been defined, and a symmetric key $\hat{K}$ can be freely invented (and then be used to decrypt $\hat{\psi}$). Note that in the latter case, care must be taken that once $\mathcal{A}$ makes an H-query $\mathsf{H}(\hat{Y}, \hat{Z})$ with $\mathrm{DDH}_{g,X}(\hat{Y}, \hat{Z}) = 1$ *later on*, then the right value $\hat{K}$ is returned.

If at any point, event F occurs, then $\mathcal{A}$ has submitted an H-query (Y, Z) for $Z = Y^x$ and effectively solved $\mathcal{B}$'s own DH challenge. This can be noticed by $\mathcal{B}$ (with the help of oracle $\mathrm{DDH}_{g,X}(\cdot,\cdot)$), and $\mathcal{B}$ can return Z. (5) follows. (A subtlety not yet mentioned is that X and Y^* are slightly differently distributed — but statistically close — in the strong DH experiment and in Game 0. This explains for the $O(2^{-\delta n(k)})$ term in (5).)

Game 2. We now change the symmetric part ψ^* of the challenge ciphertext into $\psi^* := \mathsf{E}_{K'}(R)$ for a uniform bit-string R of length $|m_0| = |m_1|$. Note that from Game 1 on, the symmetric key K' used to produce ψ^* is chosen independently. Furthermore, K' is only needed to perform decryptions of ciphertexts $\psi \neq \psi^*$ as required for ciphertexts (Y^*, ψ). Hence, we have

$$|p_2 - p_1| \leq \mathsf{Adv}^{\mathrm{cca}}_{\mathcal{B}',\mathsf{SE}}(k) \tag{6}$$

for a suitable $\mathsf{IND\text{-}CCA}$ adversary $\mathcal{B}'$ on SE.

On the other hand, $p_2 = 1/2$ since $\mathcal{A}$'s view in Game 2 is independent of b.

Putting (3,4,5,6) together yields the statement of Theorem 9.