The Twin Diffie-Hellman Problem and Applications

David Cash*, Eike Kiltz**, and Victor Shoup***

Abstract. We propose a new computational problem called the *twin Diffie-Hellman problem*. This problem is closely related to the usual (computational) Diffie-Hellman problem and can be used in many of the same cryptographic constructions that are based on the Diffie-Hellman problem. Moreover, the twin Diffie-Hellman problem is at least as hard as the ordinary Diffie-Hellman problem. However, we are able to show that the twin Diffie-Hellman problem remains hard, even in the presence of a decision oracle that recognizes solutions to the problem — this is a feature not enjoyed by the ordinary Diffie-Hellman problem. In particular, we show how to build a certain "trapdoor test" which allows us to effectively answer such decision oracle queries, without knowing any of the corresponding discrete logarithms. Our new techniques have many applications. As one such application, we present a new variant of ElGamal encryption with very short ciphertexts, and with a very simple and tight security proof, in the random oracle model, under the assumption that the ordinary Diffie-Hellman problem is hard. We present several other applications as well, including: a new variant of Diffie and Hellman's non-interactive key exchange protocol; a new variant of Cramer-Shoup encryption, with a very simple proof in the standard model; a new variant of Boneh-Franklin identity-based encryption, with very short ciphertexts; a more robust version of a password-authenticated key exchange protocol of Abdalla and Pointcheval.

1 Introduction

In some situations, basing security proofs on the hardness of the Diffie-Hellman problem is hindered by the fact that recognizing correct solutions is also apparently hard (indeed, the hardness of the latter problem is the Decisional Diffie-Hellman assumption). There are a number of ways for circumventing these technical difficulties. One way is to simply make a stronger assumption, namely, that the Diffie-Hellman problem remains hard, even given access to a corresponding

* College of Computing, Georgia Institute of Technology, USA, `cdc@gatech.edu`. Part of this work completed while at CWI.

** Cryptology & Information Security Group, CWI Amsterdam, The Netherlands, `kiltz@cwi.nl`. Supported by the research program Sentinels.

*** Dept. of Computer Science, New York University, Courant Institute, 251 Mercer Street, New York, NY 10012, USA. `shoup@cs.nyu.edu`. Supported by NSF award number CNS-0716690.

decision oracle. Another way is to work with groups that are equipped with efficient pairings, so that such a decision oracle is immediately available. However, we would like to avoid making stronger assumptions, or working with specialized groups, if at all possible.

In this paper, we introduce a new problem, the *twin Diffie Hellman problem*, which has the following interesting properties:

– the twin Diffie-Hellman problem can easily be employed in many cryptographic constructions where one would usually use the ordinary Diffie-Hellman problem, without imposing a terrible efficiency penalty;
– the twin Diffie-Hellman problem is hard, even given access to a corresponding decision oracle, assuming the ordinary Diffie-Hellman problem (without access to any oracles) is hard.

Using the twin Diffie-Hellman problem, we construct a new variant of ElGamal encryption that is secure against chosen ciphertext attack, in the random oracle model, under the assumption that the ordinary Diffie-Hellman problem is hard. Compared to other ElGamal variants with similar security properties, our scheme is attractive in that it has very short ciphertexts, and a very simple and tighter security proof.

At the heart of our method is a "trapdoor test" that allows us to implement an effective decision oracle for the twin Diffie-Hellman problem, without knowing any of the corresponding discrete logarithms. This trapdoor test has many applications, including: a new variant of Diffie and Hellman's non-interactive key exchange protocol [10], which is secure in the random oracle model assuming the Diffie-Hellman problem is hard; a new variant of Cramer-Shoup encryption [8] with a very simple security proof, in the standard model, under the *hashed* decisional Diffie-Hellman assumption; a new variant of Boneh-Franklin identity-based encryption [5], with very short ciphertexts, and a simple and tighter security proof in the random oracle model, assuming the bilinear Diffie-Hellman problem is hard; a very simple and efficient method of securing a password-authenticated key exchange protocol of Abdalla and Pointcheval [2] against server compromise, which can be proved secure, using our trapdoor test, in the random oracle model, under the Diffie-Hellman assumption.

1.1 Hashed ElGamal Encryption and its relation to the Diffie-Hellman problem

To motivate the discussion, consider the "hashed" ElGamal encryption scheme [1]. This public-key encryption scheme makes use of a group $\mathbb{G}$ of prime order q with generator $g \in \mathbb{G}$, a hash function H, and a symmetric cipher (E, D). A public key for this scheme is a random group element X, with corresponding secret key x, where $X = g^x$. To encrypt a message m, one chooses a random $y \in \mathbb{Z}_q$, computes

$$Y := g^y, \quad Z := X^y, \quad k := \mathsf{H}(Y, Z), \quad c := \mathsf{E}_k(m),$$

and the ciphertext is (Y, c). Decryption works in the obvious way: given the ciphertext (Y, c), and secret key x, one computes

$$Z := Y^x, \quad k := \mathsf{H}(Y, Z), \quad m := \mathsf{D}_k(c).$$

THE DIFFIE-HELLMAN ASSUMPTION. Clearly, the hashed ElGamal encryption scheme is secure *only if* it is hard to compute Z, given the values X and Y. Define

$$\mathrm{dh}(X, Y) := Z, \quad \text{where } X = g^x,\ Y = g^y, \text{ and } Z = g^{xy}. \tag{1}$$

The problem of computing $\mathrm{dh}(X, Y)$ given random $X, Y \in \mathbb{G}$ is the *DH problem*. The *DH assumption* asserts that this problem is hard. However, this assumption is *not* sufficient to establish the security of hashed ElGamal against a *chosen ciphertext attack*, regardless of what security properties the hash function H may enjoy.

To illustrate the problem, suppose that an adversary selects group elements $\hat{Y}$ and $\hat{Z}$ in some arbitrary way, and computes $\hat{k} := \mathsf{H}(\hat{Y}, \hat{Z})$ and $\hat{c} := \mathsf{E}_{\hat{k}}(\hat{m})$ for some arbitrary message $\hat{m}$. Further, suppose the adversary gives the ciphertext $(\hat{Y}, \hat{c})$ to a "decryption oracle," obtaining the decryption m. Now, it is very likely that $\hat{m} = m$ if and only if $\hat{Z} = \mathrm{dh}(X, \hat{Y})$. Thus, the decryption oracle can be used by the adversary as an oracle to answer questions of the form "is $\mathrm{dh}(X, \hat{Y}) = \hat{Z}$?" for group elements $\hat{Y}$ and $\hat{Z}$ of the adversary's choosing. In general, the adversary would not be able to efficiently answer such questions on his own, and so the decryption oracle is leaking some information about that secret key x which could conceivably be used to break the encryption scheme.

THE STRONG DH ASSUMPTION. Therefore, to establish the security of hashed ElGamal against chosen ciphertext attack, we need a stronger assumption. For $X, \hat{Y}, \hat{Z} \in \mathbb{G}$, define the predicate

$$\mathrm{dhp}(X, \hat{Y}, \hat{Z}) := \quad \mathrm{dh}(X, \hat{Y}) \stackrel{?}{=} \hat{Z}.$$

At a bare minimum, we need to assume that it is hard to compute $\mathrm{dh}(X, Y)$, given random $X, Y \in \mathbb{G}$, along with access to a *decision oracle* for the predicate $\mathrm{dhp}(X, \cdot, \cdot)$, which on input $(\hat{Y}, \hat{Z})$, returns $\mathrm{dhp}(X, \hat{Y}, \hat{Z})$. This assumption is called the *strong DH assumption* [1].[1] Moreover, it is not hard to prove, if H is modeled as a random oracle, that hashed ElGamal is secure against chosen ciphertext attack under the strong DH assumption, and under the assumption that the underlying symmetric cipher is itself secure against chosen ciphertext attack. This was proved in [1, 21], for a variant scheme in which Y is not included in the hash; including Y in the hash gives a more efficient security reduction (see [9]). Note that the strong DH assumption is different (and weaker) than the so called *gap DH assumption* [24] where an adversary gets access to a *full* decision oracle for the predicate $\mathrm{dhp}(\cdot, \cdot, \cdot)$, which on input $(\hat{X}, \hat{Y}, \hat{Z})$, returns $\mathrm{dhp}(\hat{X}, \hat{Y}, \hat{Z})$.

[1] We remark that in more recent papers the name strong DH assumption also sometimes refers to a different assumption defined over bilinear maps [3]. We follow the original terminology from [1].

1.2 The Twin Diffie-Hellman Assumptions

For general groups, the strong DH assumption may be strictly stronger than the DH assumption. One of the main results of this paper is to present a slightly modified version of the DH problem that is just as useful as the (ordinary) DH problem, and which is just as hard as the (ordinary) DH problem, *even given access to a corresponding decision oracle*. Using this, we get a modified version of hashed ElGamal encryption which can be proved secure under the (ordinary) DH assumption, in the random oracle model. This modified system is just a bit less efficient than the original system.

Again, let $\mathbb{G}$ be a cyclic group with generator g, and of prime order q. Let dh be defined as in (1). Define the function

$$\begin{aligned} \mathrm{2dh}: \quad & \mathbb{G}^3 \to \mathbb{G}^2 \\ & (X_1, X_2, Y) \mapsto (\mathrm{dh}(X_1, Y), \mathrm{dh}(X_2, Y)). \end{aligned}$$

We call this the *twin DH function*. One can also define a corresponding *twin DH predicate*:

$$\mathrm{2dhp}(X_1, X_2, \hat{Y}, \hat{Z}_1, \hat{Z}_2) := \quad \mathrm{2dh}(X_1, X_2, \hat{Y}) \stackrel{?}{=} (\hat{Z}_1, \hat{Z}_2).$$

The *twin DH assumption* states it is hard to compute $\mathrm{2dh}(X_1, X_2, Y)$, given random $X_1, X_2, Y \in \mathbb{G}$. It is clear that the DH assumption implies the twin DH assumption. The *strong twin DH assumption* states that it is hard to compute $\mathrm{2dh}(X_1, X_2, Y)$, given random $X_1, X_2, Y \in \mathbb{G}$, along with access to a *decision oracle* for the predicate $\mathrm{2dhp}(X_1, X_2, \cdot, \cdot, \cdot)$, which on input $(\hat{Y}, \hat{Z}_1, \hat{Z}_2)$, returns $\mathrm{2dhp}(X_1, X_2, \hat{Y}, \hat{Z}_1, \hat{Z}_2)$.

One of our main results is the following:

Theorem 1. *The (ordinary) DH assumption holds if and only if the strong twin DH assumption holds.*

The non-trivial direction to prove is that the DH assumption implies the strong twin DH assumption.

A Trapdoor Test. While Theorem 1 has direct applications, the basic tool that is used to prove the theorem, which is a kind of "trapdoor test," has even wider applications. Roughly stated, the trapdoor test works as follows: given a random group element X_1, we can efficiently construct a random group element X_2, together with a secret "trapdoor" τ, such that

- X_1 and X_2 are independent (as random variables), and
- if we are given group elements $\hat{Y}, \hat{Z}_1, \hat{Z}_2$, computed as functions of X_1 and X_2 (but not τ), then using τ, we can efficiently evaluate the predicate $\mathrm{2dhp}(X_1, X_2, \hat{Y}, \hat{Z}_1, \hat{Z}_2)$, making a mistake with only negligible probability.

We note that our trapdoor test actually appears implicitly in Shoup's DH self-corrector [28]; apparently, its implications were not understood at the time, although the techniques of Cramer and Shoup [8] are in some sense an extension of the idea. Due to space constraints we must defer the details of the connection between our trapdoor test and Shoup's DH self-corrector to the full version of this paper.

1.3 Applications and Results

The twin ElGamal encryption scheme Theorem 1 suggests the following *twin ElGamal encryption scheme*. This scheme makes use of a hash function H and a symmetric cipher (E, D). A public key for this scheme is a pair of random group elements (X_1, X_2), with corresponding secret key is (x_1, x_2), where $X_i = g^{x_i}$ for $i = 1, 2$. To encrypt a message m, one chooses a random $y \in \mathbb{Z}_q$, computes

$$Y := g^y, \quad Z_1 := X_1^y, \quad Z_2 := X_2^y, \quad k := \mathsf{H}(Y, Z_1, Z_2), \quad c := \mathsf{E}_k(m),$$

and the ciphertext is (Y, c). Decryption works in the obvious way: given the ciphertext (Y, c), and secret key (x_1, x_2), one computes

$$Z_1 := Y^{x_1}, \quad Z_2 := Y^{x_2}, \quad k := \mathsf{H}(Y, Z_1, Z_2), \quad m := \mathsf{D}_k(c).$$

The arguments in [1] and [9] trivially carry over, so that one can easily show that the twin ElGamal encryption scheme is secure against chosen ciphertext attack, under the strong twin DH assumption, and under the assumption that (E, D) is secure against chosen ciphertext attack, if H is modeled as a random oracle. Again, by Theorem 1, the same holds under the (ordinary) DH assumption.

Note that the ciphertexts for this scheme are extremely compact — no redundancy is added, as in the Fujisaki-Okamoto transformation [11]. Moreover, the security reduction for our scheme is very tight. We remark that this seems to be the first DH-based encryption scheme with short ciphertexts. All other known constructions either add redundancy to the ciphertext [11, 25, 29, 7, 18] or resort to assumptions stronger than DH [1, 9, 21].

The twin DH key-exchange protocol In their paper [10], Diffie and Hellman presented the following simple, *non-interactive* key exchange protocol. Alice chooses a random $x \in \mathbb{Z}_q$, computes $X := g^x \in \mathbb{G}$, and publishes the pair $(\mathtt{Alice}, X)$ is a public directory. Similarly, Bob chooses a random $y \in \mathbb{Z}_q$, computes $Y := g^y \in \mathbb{G}$, and publishes the pair $(\mathtt{Bob}, Y)$ in a public directory. Alice and Bob may compute the shared value $Z := g^{xy} \in \mathbb{G}$, as follows: Alice retrieves Bob's entry from the directory and computes Z as Y^x, while Bob retrieves Alice's key X, and computes Z as X^y. Before using the value Z, it is generally a good idea to hash it, together with Alice's and Bob's identities, using a cryptographic hash function H. Thus, the key that Alice and Bob actually use to encrypt data using a symmetric cipher is $k := \mathsf{H}(\mathtt{Alice}, \mathtt{Bob}, Z)$.

Unfortunately, the status of the security of this scheme is essentially the same as that of the security of hashed ElGamal against chosen ciphertext attack, if we allow an adversary to place arbitrary public keys in the public directory (without requiring some sort of "proof of possession" of a secret key).

To avoid this problem, we define the *twin DH protocol*, as follows: Alice's public key is (X_1, X_2), and her secret key is (x_1, x_2), where $X_i = g^{x_i}$ for $i = 1, 2$; similarly, Bob's public key is (Y_1, Y_2), and his secret key is (y_1, y_2), where $Y_i = g^{y_i}$

for $i = 1, 2$; their shared key is

$$k := \mathsf{H}(\mathtt{Alice}, \mathtt{Bob}, \mathrm{dh}(X_1, Y_1), \mathrm{dh}(X_1, Y_2), \mathrm{dh}(X_2, Y_1), \mathrm{dh}(X_2, Y_2)),$$

where H is a hash function. Of course, Alice computes the 4-tuple of group elements in the hash as

$$(Y_1^{x_1}, Y_2^{x_1}, Y_1^{x_2}, Y_2^{x_2}),$$

and Bob computes them as

$$(X_1^{y_1}, X_1^{y_2}, X_2^{y_1}, X_2^{y_2}).$$

Using the "trapdoor test," it is a simple matter to show that the twin DH protocol satisfies a natural and strong definition of security, under the (ordinary) DH assumption, if H is modeled as a random oracle.

A variant of Cramer-Shoup encryption We present a variant of the public-key encryption scheme by Cramer and Shoup [8]. Using our trapdoor test, along with techniques originally developed for identity-based encryption [3], we give an extremely simple proof of its security against chosen-ciphertext attack, in the standard model, under the Decisional DH assumption [12]: given X and Y, it is hard to distinguish $\mathrm{dh}(X, Y)$ from Z, for random $X, Y, Z \in \mathbb{G}$. In fact, our proof works under the weaker *hashed* Decisional DH assumption: given X and Y, it is hard to distinguish $\mathsf{H}(\mathrm{dh}(X, Y))$ from k, for random $X, Y \in \mathbb{G}$, and random k in the range of H. Note that the original Cramer-Shoup scheme cannot be proved secure under this weaker assumption — their security relies in an essential way on the Decisional DH assumption.

As a simple extension of this idea, we can obtain a new analysis of a scheme given in [17]. There, a variant of the Kurosawa-Desmedt encryption scheme is given and proved secure under the decisional DH assumption. Our analysis provides further theoretical understanding. Due to space constraints we must defer the details of this construction to the full version of this paper.

Obviously, our variants are secure under the DH assumption if H is modeled as a random oracle. We also note that by using the Goldreich-Levin theorem, a simple extension of our scheme, which is still fairly practical, can be proved secure against chosen ciphertext attack under the DH assumption.

The observation that a variant of the Cramer-Shoup encryption scheme can be proved secure under the hashed Decisional DH assumption was also made by Brent Waters, in unpublished work (personal communication, 2006) and independently by Goichiro Hanaoka and Kaoru Kurosawa, also in unpublished work [16].

Identity-based encryption Strong versions of assumptions also seem necessary to analyze some identity-based encryption (IBE) schemes that use bilinear pairings. As a further contribution, we give a twin version of the *bilinear* DH (BDH) assumption and prove that the (interactive) strong twin BDH assumption is implied by the standard BDH assumption.

The well-known IBE scheme of Boneh and Franklin [5] achieves security against chosen ciphertext, in the random oracle model, by applying the Fujisaki-Okamoto transformation. Our techniques give a different scheme with shorter ciphertexts, and a tighter security reduction. The same technique can also be applied to the scheme by Kasahara and Sakai [27] which is based on a stronger bilinear assumption but has improved efficiency.

Other applications Our twinning technique and in particular the trapdoor test can be viewed as a general framework that allows to "update" a protocol Π whose security relies on the strong DH assumption to a protocol Π' that has roughly the same complexity as Π, but whose security is solely based on the DH assumption. Apart from the applications mentioned above, we remark that this technique can also be applied to the undeniable signatures and designated confirmer signatures from [24] and the key-exchange protocols from [19].

As another application of our trapdoor test, one can easily convert the very elegant and efficient protocol of Abdalla and Pointcheval [2] for password-authenticated key exchange, into a protocol that provides security against server compromise, without adding any messages to the protocol, and still basing the security proof, in the random oracle model, on the DH assumption. For lack of space, this application will be further discussed in the full version.

2 A trapdoor test and a proof of Theorem 1

It is not hard to see that the strong twin DH implies the DH assumption. To prove that the DH implies the strong twin DH assumption, we first need our basic tool, a "trapdoor test". Its purpose will be intuitively clear in the proof of Theorem 1: in order to reduce the strong twin DH assumption to the DH assumption, the DH adversary will have to answer decision oracle queries without knowing the discrete logarithms of the elements of the strong twin DH problem instance. This tool gives us a method for doing so.

Theorem 2 (Trapdoor Test). *Let $\mathbb{G}$ be a cyclic group of prime order q, generated by $g \in \mathbb{G}$. Suppose X_1, r, s are mutually independent random variables, where X_1 takes values in $\mathbb{G}$, and each of r, s is uniformly distributed over $\mathbb{Z}_q$, and define the random variable $X_2 := g^s / X_1^r$. Further, suppose that $\hat{Y}, \hat{Z}_1, \hat{Z}_2$ are random variables taking values in $\mathbb{G}$, each of which is defined as some function of X_1 and X_2. Then we have:*

(i) X_2 is uniformly distributed over $\mathbb{G}$;
(ii) X_1 and X_2 are independent;
(iii) if $X_1 = g^{x_1}$ and $X_2 = g^{x_2}$, then the probability that the truth value of

$$\hat{Z}_1^r \hat{Z}_2 = \hat{Y}^s \tag{2}$$

does not agree with the truth value of

$$\hat{Z}_1 = \hat{Y}^{x_1} \wedge \ \hat{Z}_2 = \hat{Y}^{x_2} \tag{3}$$

is at most $1/q$; moreover, if (3) holds, then (2) certainly holds.

Proof. Observe that $s = rx_1 + x_2$. It is easy to verify that X_2 is uniformly distributed over $\mathbb{G}$, and that X_1, X_2, r are mutually independent, from which (i) and (ii) follow. To prove (iii), condition on fixed values of X_1 and X_2. In the resulting conditional probability space, r is uniformly distributed over $\mathbb{Z}_q$, while x_1, x_2, $\hat{Y}$, $\hat{Z}_1$, and $\hat{Z}_2$ are fixed. If (3) holds, then by multiplying together the two equations in (3), we see that (2) certainly holds. Conversely, if (3) does not hold, we show that (2) holds with probability at most $1/q$. Observe that (2) is equivalent to

$$(\hat{Z}_1/\hat{Y}^{x_1})^r = \hat{Y}^{x_2}/\hat{Z}_2. \tag{4}$$

It is not hard to see that if $\hat{Z}_1 = \hat{Y}^{x_1}$ and $\hat{Z}_2 \neq \hat{Y}^{x_2}$, then (4) certainly does not hold. This leaves us with the case $\hat{Z}_1 \neq \hat{Y}^{x_1}$. But in this case, the left hand side of (4) is a random element of $\mathbb{G}$ (since r is uniformly distributed over $\mathbb{Z}_q$), but the right hand side is a fixed element of $\mathbb{G}$. Thus, (4) holds with probability $1/q$ in this case.

Using this tool, we can easily prove Theorem 1. So that we can give a concrete security result, let us define some terms. For an adversary $\mathcal{B}$, let us define his *DH advantage*, denoted $\mathsf{AdvDH}_{\mathcal{B},\mathbb{G}}$, to be the probability that $\mathcal{B}$ computes $\mathrm{dh}(X, Y)$, given random $X, Y \in \mathbb{G}$. For an adversary $\mathcal{A}$, let us define his *strong twin DH advantage*, denoted $\mathsf{Adv2DH}_{\mathcal{A},\mathbb{G}}$, to be the probability that $\mathcal{A}$ computes $2\mathrm{dh}(X_1, X_2, Y)$, given random $X_1, X_2, Y \in \mathbb{G}$, along with access to a *decision oracle* for the predicate $2\mathrm{dhp}(X_1, X_2, \cdot, \cdot, \cdot)$, which on input $\hat{Y}, \hat{Z}_1, \hat{Z}_2$, returns $2\mathrm{dhp}(X_1, X_2, \hat{Y}, \hat{Z}_1, \hat{Z}_2)$.

Theorem 1 is a special case of the following:

Theorem 3. *Suppose $\mathcal{A}$ is a strong twin DH adversary that makes at most Q_{d} queries to its decision oracle, and runs in time at most τ. Then there exists a DH adversary $\mathcal{B}$ with the following properties: $\mathcal{B}$ runs in time at most τ, plus the time to perform $O(Q_{\mathrm{d}} \log q)$ group operations and some minor bookkeeping; moreover,*

$$\mathsf{Adv2DH}_{\mathcal{A},\mathbb{G}} \leq \mathsf{AdvDH}_{\mathcal{B},\mathbb{G}} + Q_{\mathrm{d}}/q.$$

In addition, if $\mathcal{B}$ does not output "failure," then its output is correct with probability at least $1 - 1/q$.

Proof. Our DH adversary $\mathcal{B}$ works as follows, given a challenge instance (X, Y) of the DH problem. First, $\mathcal{B}$ chooses $r, s \in \mathbb{Z}_q$ at random, sets $X_1 := X$ and $X_2 := g^s/X_1^r$, and gives $\mathcal{A}$ the challenge instance (X_1, X_2, Y). Second, $\mathcal{B}$ processes each decision query $(\hat{Y}, \hat{Z}_1, \hat{Z}_2)$ by testing if $\hat{Z}_1\hat{Z}_2^r = \hat{Y}^s$ holds. Finally, if and when $\mathcal{A}$ outputs (Z_1, Z_2), $\mathcal{B}$ tests if this output is correct by testing if $Z_1 Z_2^r = Y^s$ holds; if this does not hold, then $\mathcal{B}$ outputs "failure," and otherwise, $\mathcal{B}$ outputs Z_1. The proof is easily completed using Theorem 2.

3 Twin ElGamal encryption

3.1 Model and security

We recall the definition for chosen ciphertext security of a public-key encryption scheme, denoted PKE. Consider the usual chosen ciphertext attack game, played between a challenger and a adversary $\mathcal{A}$:

1. The challenger generates a public key/secret key pair, and gives the public key to $\mathcal{A}$;
2. $\mathcal{A}$ makes a number of *decryption queries* to the challenger; each such query is a ciphertext $\hat{C}$; the challenger decrypts $\hat{C}$, and sends the result to $\mathcal{A}$;
3. $\mathcal{A}$ makes one *challenge query*, which is a pair of messages (m_0, m_1); the challenger chooses $b \in \{0, 1\}$ at random, encrypts m_b, and sends the resulting ciphertext C to $\mathcal{A}$;
4. $\mathcal{A}$ makes more *decryption queries*, just as in step 2, but with the restriction that $\hat{C} \neq C$;
5. $\mathcal{A}$ outputs $\hat{b} \in \{0, 1\}$.

The advantage $\mathsf{AdvCCA}_{\mathcal{A},\mathsf{PKE}}$ is defined to be $|\Pr[\hat{b} = b] - 1/2|$. The scheme PKE is said to be secure against chosen ciphertext attack if for all efficient adversaries $\mathcal{A}$, the advantage $\mathsf{AdvCCA}_{\mathcal{A},\mathsf{PKE}}$ is negligible.

If we wish to analyze a scheme PKE in the random oracle model, then hash functions are replaced by random oracle queries as appropriate, and both challenger and adversary are given access to the random oracle in the above attack game. We write $\mathsf{AdvCCA}^{\mathrm{ro}}_{\mathcal{A},\mathsf{PKE}}$ for the corresponding advantage in the random oracle model.

If $\mathsf{SE} = (\mathsf{E}, \mathsf{D})$ is a symmetric cipher, then one defines security against chosen ciphertext attack in exactly the same way, except that in step 1 of the above attack game, the challenger simply generates a secret key and step 2 of the above attack game is left out. The advantage $\mathsf{AdvCCA}_{\mathcal{A},\mathsf{SE}}$ is defined in exactly the same way, and SE is said to be secure against chosen ciphertext attack if for all efficient adversaries $\mathcal{A}$, the advantage $\mathsf{AdvCCA}_{\mathcal{A},\mathsf{SE}}$ is negligible.

The usual construction of a chosen-ciphertext secure symmetric encryption scheme is to combine a one-time pad and a message-authentication code (MAC). We remark that such schemes do not necessarily add any redundancy to the symmetric ciphertext. In fact, Phan and Pointcheval [26] showed that a *strong PRP* [13] directly implies a length-preserving chosen-ciphertext secure symmetric encryption scheme that avoids the usual overhead due to the MAC. In practice one can use certain modes of operation (e.g., CMC [15]) to encrypt large messages. The resulting scheme is chosen-ciphertext secure provided that the underlying block-cipher is a strong PRP.

3.2 Security of the Twin ElGamal scheme

We are now able to establish the security of the twin ElGamal encryption scheme described in §1.3, which we denote $\mathsf{PKE}_{\mathrm{2dh}}$. The security will be based on the strong twin DH assumption, of course, and this allows us to borrow the "oracle patching" technique from previous analyses of hashed ElGamal encryption based on the strong DH assumption. We stress, however, that unlike previous applications of this technique, the end result is a scheme based on the original DH assumption.

Theorem 4. *Suppose* H *is modeled as a random oracle and that the DH assumption holds. Then* $\mathsf{PKE}_{2\mathrm{dh}}$ *is secure against chosen ciphertext attack.*

In particular, suppose $\mathcal{A}$ *is an adversary that carries out a chosen ciphertext attack against* $\mathsf{PKE}_{2\mathrm{dh}}$ *in the random oracle model, and that* $\mathcal{A}$ *runs in time* τ*, and makes at most* Q_{h} *hash queries and* Q_{d} *decryption queries. Then there exists a DH adversary* $\mathcal{B}_{\mathrm{dh}}$ *and an adversary* $\mathcal{B}_{\mathrm{sym}}$ *that carries out a chosen ciphertext attack against* SE*, such that both* $\mathcal{B}_{\mathrm{dh}}$ *and* $\mathcal{B}_{\mathrm{sym}}$ *run in time at most* τ*, plus the time to perform* $O((Q_{\mathrm{h}} + Q_{\mathrm{d}}) \log q)$ *group operations; moreover,*

$$\mathsf{AdvCCA}^{\mathrm{ro}}_{\mathcal{A},\mathsf{PKE}_{2\mathrm{dh}}} \leq \mathsf{AdvDH}_{\mathcal{B}_{\mathrm{dh}},\mathbb{G}} + \mathsf{AdvCCA}_{\mathcal{B}_{\mathrm{sym}},\mathsf{SE}} + Q_{\mathrm{h}}/q.$$

Given the equivalence between the strong 2DH and the DH assumption from Theorem 1, the proof of Theorem 4 is quite standard, but must be deferred to the full version.

Instantiating $\mathsf{PKE}_{2\mathrm{dh}}$ with a length-preserving chosen-ciphertext secure symmetric encryption scheme, we obtain a DH-based chosen-ciphertext secure encryption scheme with the following properties.

Optimal ciphertext overhead. The ciphertext overhead, i.e. ciphertext size minus plaintext size, is exactly one group element, which is optimal for Diffie-Hellman based schemes. For concreteness, for $\kappa = 128$ bit security, a typical implementation in elliptic curve groups gives a concrete ciphertext overhead of 256 bits.

Encryption/decryption efficiency. Encryption needs three exponentiations in $\mathbb{G}$, one of which is to the fixed-base g (that can be shared among many public-keys). Decryption only needs one sequential exponentiation in $\mathbb{G}$ to compute Y^{x_1} and Y^{x_2} simultaneously, which is nearly as efficient as one single exponentiation (see, e.g., [23]).

4 Non-interactive key exchange

In this section we give a model and security definition for non-interactive key exchange and then analyze the twin DH protocol from section §1.3. Strangely, after the seminal work of Diffie and Hellman on this subject, it does not seem to have been explored further in the literature, except in the identity-based setting.

4.1 Model and security

A non-interactive key exchange scheme KE consists of two algorithms: one for key generation and one for computing paired keys. The key generation algorithm is probabilistic and outputs a public key/secret key pair. The paired key algorithm takes as input an identity and public key along with another identity and a secret key, and outputs a shared key for the two identities. Here, identities are arbitrary strings chosen by the users, and the key authority does not generate keys itself but acts only as a phonebook.

For security we define an experiment between a challenger and an adversary $\mathcal{A}$. In this experiment, the challenger takes a random bit b as input and answers oracle queries for $\mathcal{A}$ until $\mathcal{A}$ outputs a bit $\hat{b}$. The challenger answers the following types of queries for $\mathcal{A}$:

Register honest user ID. $\mathcal{A}$ supplies a string id. The challenger runs the key generation algorithm to generate a public key/secret key pair $(\mathrm{pk}, \mathrm{sk})$ and records the tuple $(\mathrm{honest}, id, \mathrm{pk}, \mathrm{sk})$ for later. The challenger returns pk to $\mathcal{A}$.

Register corrupt user ID. In this type of query, $\mathcal{A}$ supplies both the string id and a public key pk. The challenger records the tuple $(\mathrm{corrupt}, id, \mathrm{pk})$ for later.

Get honest paired key. Here $\mathcal{A}$ supplies two identities id, id' that were registered as honest users. Now the challenger uses the bit b: if $b = 0$, the challenger runs the paired key algorithm using the public key for id and the secret key for id'. If $b = 1$, the challenger generates a random key, records it for later, and returns that to the adversary. To keep things consistent, the challenger returns the same random key for the set $\{id, id'\}$ every time $\mathcal{A}$ queries for their paired key (perhaps in reversed order).

Get corrupt paired key. Here $\mathcal{A}$ supplies two identities id, id', where id was registered as corrupt and id' was registered as honest. The challenger runs the paired key algorithm using the public key for id and the secret key for id' and returns the paired key.

When the adversary finally outputs $\hat{b}$, it wins the experiment if $\hat{b} = b$. For an adversary $\mathcal{A}$, we define its advantage $\mathsf{AdvKA}_{\mathcal{A},\mathsf{KE}}$ in this experiment to be $|\Pr[\hat{b} = b] - 1/2|$. When a hash function is modeled as a random oracle in the experiment, we denote the adversary's advantage by $\mathsf{AdvKA}^{\mathrm{ro}}_{\mathcal{A},\mathsf{KE}}$. We say that a non-interactive key-exchange scheme KE is *secure against active attacks* if for all efficient adversaries $\mathcal{A}$, the advantage $\mathsf{AdvKA}^{\mathrm{ro}}_{\mathcal{A},\mathsf{KE}}$ is negligible.

We note that in the ideal version of the experiment above (when $b = 1$), the challenger returns the same random key for the honest paired key queries for (id, id') and (id', id). This essentially means that there should be no concept of "roles" in the model and that protocols should implement something like a canonical ordering of all the identities to implicitly define roles if needed.

4.2 Security of the twin DH protocol

As stated above, we can prove the twin DH protocol secure under the DH assumption using our trapdoor test. We denote the twin DH protocol by $\mathsf{KA}_{\mathrm{2dh}}$. A complete proof will be given in the full version.

Theorem 5. *Suppose* H *is modeled as a random oracle and that the DH assumption holds. Then* $\mathsf{KA}_{2\mathrm{dh}}$ *is secure against active attacks.*

In particular, suppose $\mathcal{A}$ *is an adversary that attacks* $\mathsf{KA}_{2\mathrm{dh}}$ *in the random oracle model, and that* $\mathcal{A}$ *runs in time* τ*, and makes at most a total of* Q *oracle queries of all types. Then there exists a DH adversary* $\mathcal{B}_{\mathrm{dh}}$ *that runs in time at most* τ *plus the time to perform* $O(Q \log q)$ *group operations; moreover,*

$$\mathsf{AdvKA}^{\mathrm{ro}}_{\mathcal{A},\mathsf{KA}_{2\mathrm{dh}}} \leq 2\mathsf{AdvDH}_{\mathcal{B}_{\mathrm{dh}},\mathbb{G}} + 4Q/q.$$

5 A variant of the Cramer-Shoup encryption scheme

5.1 The (Twin) DDH assumption

Let $\mathbb{G}$ be a group of order q and let g be a random generator. Distinguishing the two distributions $(X, Y, \mathrm{dh}(X, Y))$ and (X, Y, Z) for random $X, Y, Z \in \mathbb{G}$ is the *Decision Diffie-Hellman* (DDH) problem. The *DDH assumption* states that the DDH problem is hard. As a natural decision variant of the Twin DH problem, the *Twin DDH problem* is distinguishing the two distributions $(X_1, X_2, Y, \mathrm{dh}(X_1, Y))$ and (X_1, X_2, Y, Z) for random $X_1, X_2, Y, Z \in \mathbb{G}$. The *Strong Twin DDH assumption* states that the Twin DDH problem is hard, even given access to a decision oracle for the predicate for $2\mathrm{dhp}(X_1, X_2, \cdot, \cdot, \cdot)$, which on input $(\hat{Y}, \hat{Z}_1, \hat{Z}_2)$ returns $2\mathrm{dhp}(X_1, X_2, \hat{Y}, \hat{Z}_1, \hat{Z}_2)$. (Note the value $\mathrm{dh}(X_2, Y)$ is never provided as input to the distinguisher since otherwise the Strong Twin DDH assumption could trivially broken using the 2dhp oracle.)

We also consider a potentially weaker "hashed variants" of the above two assumptions. For a hash function $\mathsf{H} : \mathbb{G} \to \{0, 1\}^{\kappa}$, the *Hashed DDH* problem is to distinguish the two distributions $(X, Y, \mathsf{H}(\mathrm{dh}(X, Y))$ and (X, Y, k), for random $X, Y \in \mathbb{G}$ and $k \in \{0, 1\}^{\kappa}$. The *Hashed DDH assumption* states that the Hashed DDH problem is hard. In the same way, we can consider the Strong Twin Hashed DDH assumption.

We stress that the (Strong Twin) Hashed DDH assumption simplifies to the (Strong Twin) DDH assumption when H is the identity. Furthermore, there are natural groups (such as non-prime-order groups) where the DDH problem is known to be easy yet the Hashed DDH problem is still assumed to be hard for a reasonable choice of the hash function [12]. If H is modeled as random oracle then the Hashed DDH and the DH assumption become equivalent.

Using the trapdoor test in Theorem 2, we can prove an analog of Theorem 3.

Theorem 6. *The (Hashed) DDH assumption holds if and only if the Strong Twin (Hashed) DDH assumption holds. In particular, suppose* $\mathcal{A}$ *is a Strong Twin (Hashed) DDH adversary that makes at most* Q_{d} *queries to its decision oracle, and runs in time at most* τ*. Then there exists a (Hashed) DDH adversary* $\mathcal{B}$ *with the following properties:* $\mathcal{B}$ *runs in time at most* τ*, plus the time to perform* $O(Q_{\mathrm{d}} \log q)$ *group operations and some minor bookkeeping; moreover,*

$$\mathsf{Adv2DDH}_{\mathcal{A},\mathbb{G}} \leq \mathsf{AdvDDH}_{\mathcal{B},\mathbb{G}} + Q_{\mathrm{d}}/q.$$

5.2 A variant of the Cramer-Shoup scheme

We now can consider the following encryption scheme which we call $\mathsf{PKE}_{\widetilde{\mathrm{cs}}}$. This scheme makes use of a symmetric cipher (E, D) and a hash function $\mathsf{T} : \mathbb{G} \to \mathbb{Z}_q$ which we assume to be target collision-resistant [9]. A public key for this scheme is a tuple of random group elements $(X_1, \tilde{X}_1, X_2, \tilde{X}_2)$, with corresponding secret key $(x_1, \tilde{x}_1, x_2, \tilde{x}_2)$, where $X_i = g^{x_i}$ and $\tilde{X}_i = g^{\tilde{x}_i}$ for $i = 1, 2$. To encrypt a message m, one chooses a random $y \in \mathbb{Z}_q$, computes

$$Y := g^y,\ t := \mathsf{T}(Y),\ Z_1 := (X_1^t \tilde{X}_1)^y,\ Z_2 := (X_2^t \tilde{X}_2)^y,\ k := \mathsf{H}(X_1^y),\ c := \mathsf{E}_k(m),$$

and the ciphertext is (Y, Z_1, Z_2, c). Decryption works as follows: given the ciphertext (Y, Z_1, Z_2, c), and secret key $(x_1, \tilde{x}_1, x_2, \tilde{x}_2)$, one computes $t := \mathsf{T}(Y)$ and checks if

$$Y^{x_1 t + \tilde{x}_1} = Z_1 \text{ and } Y^{x_2 t + \tilde{x}_2} = Z_2. \tag{5}$$

If not (we say the ciphertext is *not consistent*), reject; otherwise, compute

$$k := \mathsf{H}(Y^{x_1}), \quad m := \mathsf{D}_k(c).$$

We remark that since $|\mathbb{G}| = |\mathbb{Z}_q| = q$, hash function T could be a bijection. See [6] for efficient constructions for certain groups $\mathbb{G}$.

Relation to Cramer-Shoup. Our scheme is very similar to the one by Cramer and Shoup [8]. Syntactically, the difference is that in Cramer-Shoup the value Z_1 is computed as $Z_1 = X_3^y$ (where X_3 is another random group element in the public key) and t is computed as $t = \mathsf{T}(Y, Z_1)$. However, our variant allows for a *simple security proof* based on the *Hashed* DDH assumption whereas for the Cramer-Shoup scheme only a proof based on the DDH assumption is known (and the currently known proofs do not allow for it).

5.3 Security

We now show that, using the trapdoor test, $\mathsf{PKE}_{\widetilde{\mathrm{cs}}}$ allows for a very elementary proof under the Hashed DDH assumption. We stress that are security proof is not in the random oracle model.

Theorem 7. *Suppose T is a target collision resistant hash function. Further, suppose the Hashed DDH assumption holds, and that the symmetric cipher $\mathsf{SE} = (\mathsf{E}, \mathsf{D})$ is secure against chosen ciphertext attack. Then $\mathsf{PKE}_{\widetilde{\mathrm{cs}}}$ is secure against chosen ciphertext attack.*

In particular, suppose $\mathcal{A}$ is an adversary that carries out a chosen ciphertext attack against $\mathsf{PKE}_{\widetilde{\mathrm{cs}}}$ and that $\mathcal{A}$ runs in time τ, and makes at most Q_{d} decryption queries. Then there exists a Hashed DDH adversary $\mathcal{B}_{\mathrm{ddh}}$, an adversary $\mathcal{B}_{\mathrm{sym}}$ that carries out a chosen ciphertext attack against SE, and a TCR adversary $\mathcal{B}_{\mathrm{tcr}}$ such that both $\mathcal{B}_{\mathrm{ddh}}$, $\mathcal{B}_{\mathrm{sym}}$ and $\mathcal{B}_{\mathrm{tcr}}$ run in time at most τ, plus the time to perform $O(Q_{\mathrm{d}} \log q)$ group operations; moreover,

$$\mathsf{AdvCCA}_{\mathcal{A},\mathsf{PKE}_{\widetilde{\mathrm{cs}}}} \leq \mathsf{AdvDDH}_{\mathcal{B}_{\mathrm{ddh}},\mathbb{G},\mathsf{H}} + \mathsf{AdvCCA}_{\mathcal{B}_{\mathrm{sym}},\mathsf{SE}} + \mathsf{AdvTCR}_{\mathcal{B}_{\mathrm{tcr}},\mathsf{T}} + Q_{\mathrm{d}}/q.$$

Proof. We proceed with a sequence of games.

Game 0. Let Game 0 be the original chosen ciphertext attack game, and let S_0 be the event that $\hat{b} = b$ in this game.

$$\mathsf{AdvCCA}_{\mathcal{A},\mathsf{PKE}_{\widetilde{\mathrm{cs}}}} = |\Pr[S_0] - 1/2|. \tag{6}$$

Game 1 Let Game 1 be like Game 0, but with the following difference. Game 1 aborts if the adversary, at any time, makes a decryption query containing a $\hat{Y}$ such that $\hat{Y} \neq Y$ and $\mathsf{T}(\hat{Y}) = \mathsf{T}(Y)$ where Y comes from the challenge ciphertext. Using a standard argument from [9] it is easy to show that

$$|\Pr[S_1] - \Pr[S_0]| \leq \mathsf{AdvTCR}_{\mathcal{B}_{\mathrm{tcr}},\mathsf{T}}. \tag{7}$$

Game 2. Let Game 2 be as Game 1 with the following differences. For computing the public-key the experiment picks $x_1, x_2, y, a_1, a_2 \in \mathbb{Z}_q$ at random and computes $X_1 = g^{x_1}$, $X_2 = g^{x_2}$, and $Y = g^y$. Next, it computes $t := \mathsf{T}(Y)$ and

$$\tilde{X}_1 := X_1^{-t} g^{a_1}, \quad \tilde{X}_2 := X_2^{-t} g^{a_2}.$$

Note that the way the public-key is setup uses a technique to prove selective-ID security for IBE schemes [3].

The challenge ciphertext (Y, Z_1, Z_2, c) for message m_b is computed as

$$t := \mathsf{T}(Y), \quad Z_1 := Y^{a_1}, \quad Z_2 := Y^{a_2}, \quad k := \mathsf{H}(X_1^y), \quad c := \mathsf{E}_k(m_b). \tag{8}$$

This is a correctly distributed ciphertext for m_b and randomness $y = \log_g(Y)$ since, for $i = 1, 2$, $(X_i^t \tilde{X}_i)^y = (X_i^{t-t} g^{a_i})^y = (g^{a_i})^y = Y^{a_i} = Z_i$. We can assume (Y, Z_1, Z_2, k) to be computed in the beginning of the experiment since they are independent of m_0, m_1.

A decryption query for ciphertext $(\hat{Y}, \hat{Z}_1, \hat{Z}_2, \hat{c})$ is answered as follows. Compute $\hat{t} = \mathsf{T}(\hat{Y})$. If $t = \hat{t}$ then verify consistency by checking if $Z_1 = \hat{Z}_1$ and $Z_2 = \hat{Z}_2$. If the ciphertext is consistent then use the challenge key k defined in (8) to decrypt $\hat{c}$. If $t \neq \hat{t}$ then proceed as follows. For $i = 1, 2$, compute $\bar{Z}_i = (\hat{Z}_i / \hat{Y}^{a_i})^{1/(\hat{t}-t)}$. Consistency of the ciphertext is verified by checking if

$$\hat{Y}^{x_1} = \bar{Z}_1 \text{ and } \hat{Y}^{x_2} = \bar{Z}_2. \tag{9}$$

Let $\hat{y} = \log_g \hat{Y}$. The value $\hat{Z}_i$ was correctly generated iff $\hat{Z}_i = (X_i^{\hat{t}} \tilde{X}_i)^{\hat{y}} = (X_i^{\hat{t}-t} g^{a_i})^{\hat{y}} = (\hat{Y}^{x_i})^{\hat{t}-t} \cdot \hat{Y}^{a_i}$ which is equivalent to $\bar{Z}_i = \hat{Y}^{x_i}$. Hence, (9) is equivalent to the test from the original scheme (5). If the ciphertext is consistent then one can use the symmetric key $\hat{k} = \mathsf{H}(\bar{Z}_1) = \mathsf{H}(\hat{Y}^{x_1})$ to decrypt $\hat{c}$ and return $\hat{m} = \mathsf{D}_{\hat{k}}(\hat{c})$.

Let S_2 be the event that $\hat{b} = b$ in this game. As we have seen,

$$\Pr[S_2] = \Pr[S_1]. \tag{10}$$

Game 3. Let Game 3 be as Game 2 with the only difference that the value k to compute that challenge ciphertext is now chosen at random from $\mathbb{G}$. We claim that

$$|\Pr[S_3] - \Pr[S_2]| \leq \mathsf{Adv2DDH}_{\mathcal{B}_{\mathrm{2ddh}},\mathbb{G},\mathsf{H}}, \tag{11}$$

where $\mathcal{B}_{\mathrm{2ddh}}$ is an efficient Strong Twin Hashed DDH adversary that makes at most Q_{d} queries to the decision oracle. $\mathcal{B}_{\mathrm{2ddh}}$ is defined as follows. Using the values (X_1, X_2, Y, k) from its challenge (where either $k = \mathsf{H}(\mathrm{dh}(X_1, Y))$ or k is random), adversary $\mathcal{B}_{\mathrm{2ddh}}$ runs (without knowing x_1, x_2, y) the experiment as described in Game 2 using k as the challenge key in (8) to encrypt m_b. Note that the only point where Games 2 and 3 make use of x_1 and x_2 is the consistency check (9) which $\mathcal{B}_{\mathrm{2ddh}}$ equivalently implements using the 2dhp oracle, i.e. by checking if

$$\mathrm{2dhp}(X_1, X_2, \hat{Y}, \bar{Z}_1, \bar{Z}_2)$$

holds. We have that if $k = \mathsf{H}(\mathrm{dh}(X_1, Y)) \in \{0,1\}^\kappa$, this perfectly simulates Game 2, whereas if $k \in \{0,1\}^\kappa$ is random this perfectly simulates Game 3. This proves (11).

Finally, it is easy to see that in Game 3, the adversary is essentially playing the chosen ciphertext attack game against SE. Thus, there is an efficient adversary $\mathcal{B}_{\mathrm{sym}}$ such that

$$|\Pr[S_3] - 1/2| = \mathsf{AdvCCA}_{\mathcal{B}_{\mathrm{sym}},\mathsf{SE}}. \tag{12}$$

The theorem now follows by combining (6)–(12) with Theorem 6.

5.4 A variant with security from the DH assumption

We now consider a slight variant of the scheme $\mathsf{PKE}_{\widetilde{\mathrm{cs}}}$ that uses the Goldreich-Levin bit [14, 13] to achieve security based on the (standard) DH assumption.

Let $\nu = O(\log \kappa)$ be some integer that divides the security parameter κ and set $\ell = \kappa/\nu$. Let the public key now contain the $2\ell + 3$ group elements Y and $X_i = g^{x_i}, \tilde{X}_i = g^{\tilde{x}_i}$, for $i = 1, \ldots, \ell + 1$. Furthermore, it contains a sufficiently large random bit-strings R to extract the Diffie-Hellman hard-core bits (a string of length $\ell \cdot 2\kappa$ is sufficient). To encrypt a message m, one chooses a random $y \in \mathbb{Z}_q$, computes $Y := g^y$ and $Z_i := (X_i^t \tilde{X}_i)^y$, for $i = 1, \ldots, \ell + 1$, where $t = \mathsf{T}(Y)$. As before, the function of $Z_{\ell+1}$ is the consistency check. From each of the ℓ unique Diffie-Hellman keys $k_i = \mathsf{H}(X_i^y) \in \{0,1\}^\kappa$ $(i = 1, \ldots, \ell)$ and parts of R we can now extract a $\nu = \kappa/\ell$ simultaneous hard-core bits $k_i' \in \{0,1\}^\nu$. Finally, a concatenation of all k_i' yields a k-bit symmetric key $k \in \{0,1\}^\kappa$ that is used to encrypt m as $c = \mathsf{E}_k(m)$. The ciphertext is $(Y, Z_1, \ldots, Z_{\ell+1}, c)$. Decryption first verifies consistency of $(Y, Z_1, \ldots, Z_{\ell+1})$ by checking if $Y^{x_i t + \tilde{x}_i} = Z_i$, for all $i = 1, \ldots, \ell + 1$. Then the key k is reconstructed from the unique Diffie-Hellman keys $k_i = \mathsf{H}(Y^{x_i})$ as in encryption.

For concreteness we can consider a security parameter of $\kappa = 128$ bits and set $\nu = \log_2(\kappa) = 7$, which means the ciphertext overhead consists of $128/7 + 2 \approx 20$

group elements which account for $20 \cdot 256 \approx 5000$ bits when implemented on elliptic curves. Note that this is less than two standard RSA moduli for the same security level (3072 bits each, for $\kappa = 128$).

In the full version we show that the above scheme is chosen-ciphertext secure under the DH assumption. The proof uses a hybrid argument in connection with the trapdoor test from Theorem 2. Furthermore, it uses the Goldreich-Levin construction to extract $\nu = O(\log(\kappa))$ hard-core bits out of each Diffie-Hellman key. The security reduction is polynomial-time but due to the generic hard-core construction it is not very tight.

6 Identity Based Encryption

In this section we show how to apply the trapdoor test in Theorem 2 to identity-based encryption in pairing groups. We give a bilinear version of the strong twin DH problem and show that it can be reduced to the standard bilinear DH problem. We then use this assumption to construct a new IBE scheme that we call twin Boneh-Franklin below. The end result is a chosen ciphertext secure IBE scheme based on bilinear DH with one group element of overhead in the ciphertexts and a tighter reduction than the original scheme on which it is based.

6.1 A new bilinear assumption

In groups equipped with a pairing $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, we can define the function

$$\mathrm{bdh}(X, Y, W) := Z, \quad \text{where } X = g^x,\ Y = g^y,\ W = g^w, \text{ and } Z = \hat{e}(g, g)^{wxy}.$$

Computing $\mathrm{bdh}(X, Y, W)$ using random $X, Y, W \in \mathbb{G}$ is the *bilinear DH (or BDH) problem*. The *BDH assumption* states that computing the BDH problem is hard. We define a predicate

$$\mathrm{bdhp}(X, \hat{Y}, \hat{W}, \hat{Z}) := \quad \mathrm{bdh}(X, \hat{Y}, \hat{W}) \stackrel{?}{=} \hat{Z}.$$

We can also consider the BDH problem where, in addition to random (X, Y, W), one is also given access to an oracle that on input $(\hat{Y}, \hat{W}, \hat{Z})$ returns $\mathrm{bdhp}(X, \hat{Y}, \hat{W}, \hat{Z})$. The *strong BDH assumption* [22] states that the BDH problem remains hard even with the help of the oracle.

For reasons similar to the issue with hashed ElGamal encryption, the strong BDH assumption seems necessary to prove the CCA security of the basic version [22] of the original Boneh-Franklin IBE [5]. We can repeat the above idea and define the *twin BDH problem*, where one must compute $2\mathrm{bdh}(X_1, X_2, Y, W)$ for random X_1, X_2, Y, W, where we define

$$2\mathrm{bdh}(X_1, X_2, Y, W) := (\mathrm{bdh}(X_1, Y, W), \mathrm{bdh}(X_2, Y, W)).$$

Continuing as above, the *strong twin BDH problem* is the same as the twin BDH problem but with a suitably defined decision oracle. In this case define the predicate

$$2\mathrm{bdhp}(X_1, X_2, \hat{Y}, \hat{W}, \hat{Z}_1, \hat{Z}_2) := \quad 2\mathrm{bdh}(X_1, X_2, \hat{Y}, \hat{W}) \stackrel{?}{=} (\hat{Z}_1, \hat{Z}_2),$$

and the decision oracle takes input $(\hat{Y}, \hat{W}, \hat{Z}_1, \hat{Z}_2)$ and returns $\mathrm{2bdhp}(X_1, X_2, \hat{Y}, \hat{W}, \hat{Z}_1, \hat{Z}_2)$. The *strong twin BDH assumption* states that the BDH problem is still hard, even with access to the decision oracle.

Finally, we will need a slight generalization of the trapdoor test in Theorem 2. It is easy to check that the theorem is still true if the elements $\hat{Y}, \hat{Z}_1, \hat{Z}_2$ are in a *different* cyclic group of the same order (we will take them in the range group of the pairing). With this observation, we can prove an analog of Theorem 3.

Theorem 8. *Suppose $\mathcal{A}$ is a strong twin BDH adversary that makes at most Q_d queries to its decision oracle, and runs in time at most τ. Then there exists a BDH adversary $\mathcal{B}$ with the following properties: $\mathcal{B}$ runs in time at most τ, plus the time to perform $O(Q_\mathrm{d} \log q)$ group operations and some minor bookkeeping; moreover,*

$$\mathsf{Adv2BDH}_{\mathcal{A},\mathbb{G}} \leq \mathsf{AdvBDH}_{\mathcal{B},\mathbb{G}} + Q_\mathrm{d}/q.$$

In addition, if $\mathcal{B}$ does not output "failure," then its output is correct with probability at least $1 - 1/q$.

6.2 Twin Boneh-Franklin

For model and security definitions of IBE we refer the reader to [5]. Theorem 8 admits a simple analysis of the following IBE scheme, which we call the *twin Boneh-Franklin IBE scheme*. This scheme will use two hash functions, H (which outputs symmetric keys) and G (which outputs group elements), and a symmetric cipher (E, D). A system public key is a pair of group elements (X_1, X_2), where $X_i = g^{x_i}$ for $i = 1, 2$. The system private key is (x_1, x_2), which are selected at random from $\mathbb{Z}_q$ by the setup algorithm. The secret key for an identity $id \in \{0,1\}^*$ is $(S_1, S_2) = (\mathsf{G}(id)^{x_1}, \mathsf{G}(id)^{x_2})$. To encrypt a message m for identity id, one chooses $y \in \mathbb{Z}_q$ and random and sets

$$\begin{aligned} &Y := g^y, \quad Z_1 := \hat{e}(\mathsf{G}(id), X_1)^y, \quad Z_2 := \hat{e}(\mathsf{G}(id), X_2)^y, \\ &k := \mathsf{H}(id, Y, Z_1, Z_2), \quad c := \mathsf{E}_k(m). \end{aligned}$$

The ciphertext is (Y, c). To decrypt using the secret key (S_1, S_2) for id, one computes

$$Z_1 := \hat{e}(S_1, Y), \quad Z_2 := \hat{e}(S_2, Y), \quad k := \mathsf{H}(id, Y, Z_1, Z_2), \quad m := \mathsf{D}_k(c).$$

We shall denote this scheme $\mathsf{IBE}_{\mathrm{2dh}}$. Now we can essentially borrow the analysis of the original Boneh-Franklin scheme under the strong BDH assumption [22], except now we prove that the scheme is secure against *chosen ciphertext attack* under the strong twin BDH assumption. By Theorem 8, we get that the above IBE scheme is CCA secure under the BDH assumption if the symmetric cipher is secure and the hash functions are treated as random oracles. The security reduction here enjoys the same tightness as the reduction given in [22], which is tighter than the original analysis of the Boneh-Franklin scheme. Again, for space reasons we will give a complete statement of this result and the corresponding proof (which is mostly standard) in the full version.

We remark that our ideas can also be applied to the IBE scheme from Sakai-Kasahara [27]. The resulting IBE scheme is more efficient but its security can only be proved based on the q-BDHI assumption [4].

Acknowledgments

We thank Masayuki Abe and Tatsuaki Okamoto for interesting discussions.

References

1. Michel Abdalla, Mihir Bellare, and Phillip Rogaway. The oracle Diffie-Hellman assumptions and an analysis of DHIES. In David Naccache, editor, *CT-RSA 2001*, volume 2020 of *LNCS*, pages 143–158. Springer-Verlag, 2001.
2. Michel Abdalla and David Pointcheval. Simple password-based encrypted key exchange protocols. In Alfred Menezes, editor, *CT-RSA 2005*, volume 3376 of *LNCS*, pages 191–208. Springer-Verlag, 2005.
3. Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity based encryption without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–238. Springer-Verlag, 2004.
4. Dan Boneh and Xavier Boyen. Short signatures without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 56–73. Springer-Verlag, 2004.
5. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer-Verlag, 2001.
6. Xavier Boyen, Qixiang Mei, and Brent Waters. Direct chosen ciphertext security from identity-based techniques. In *ACM CCS 05*, pages 320–329. ACM Press, November 2005.
7. Jean-Sébastien Coron, Helena Handschuh, Marc Joye, Pascal Paillier, David Pointcheval, and Christophe Tymen. GEM: A generic chosen-ciphertext secure encryption method. In Bart Preneel, editor, *CT-RSA 2002*, volume 2271 of *LNCS*, pages 263–276. Springer-Verlag, 2002.
8. Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 13–25. Springer-Verlag, 1998.
9. Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003.
10. Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.
11. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 537–554. Springer-Verlag, 1999.
12. Rosario Gennaro, Hugo Krawczyk, and Tal Rabin. Secure Hashed Diffie-Hellman over non-DDH groups. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 361–381. Springer-Verlag, 2004.
13. Oded Goldreich. *Foundations of Cryptography: Basic Tools*, volume 1. Cambridge University Press, Cambridge, UK, 2001.

14. Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *21st ACM STOC*, pages 25–32. ACM Press, May 1989.
15. Shai Halevi and Phillip Rogaway. A tweakable enciphering mode. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 482–499. Springer-Verlag, 2003.
16. Goichiro Hanaoka and Kaoru Kurosawa. Efficient chosen ciphertext secure public key encryption under the computational Diffie-Hellman assumption. Unpublished manuscript, 2008.
17. Dennis Hofheinz and Eike Kiltz. Secure hybrid encryption from weakened key encapsulation. In Alfred Menezes, editor, *Advances in Cryptology, Proceedings of CRYPTO 2007*, *LNCS*, pages 553–571. Springer-Verlag, 2007.
18. Kwangjo Kim Joonsang Baek, Byoungcheon Lee. Secure length-saving ElGamal encryption under the computational Diffie-Hellman assumption. In *ACISP 2000*, *LNCS*, pages 49–58, Springer-Verlag, 2000.
19. Caroline Kudla and Kenneth G. Paterson. Modular security proofs for key agreement protocols. In Bimal K. Roy, editor, *ASIACRYPT 2005*, volume 3788 of *LNCS*, pages 549–565. Springer-Verlag, 2005.
20. Kaoru Kurosawa and Yvo Desmedt. A new paradigm of hybrid encryption scheme. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 426–442. Springer-Verlag, 2004.
21. Kaoru Kurosawa and Toshihiko Matsuo. How to remove MAC from DHIES. In *ACISP 2004*, *PNCS*, pages 236–247. Springer-Verlag, 2004.
22. Benoît Libert and Jean-Jacques Quisquater. Identity based encryption without redundancy. In John Ioannidis, Angelos Keromytis, and Moti Yung, editors, *ACNS 05*, volume 3531 of *LNCS*, pages 285–300. Springer-Verlag, 2005.
23. Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography.* The CRC Press series on discrete mathematics and its applications. CRC Press, 1997.
24. Tatsuaki Okamoto and David Pointcheval. The gap-problems: A new class of problems for the security of cryptographic schemes. In Kwangjo Kim, editor, *PKC 2001*, volume 1992 of *LNCS*, pages 104–118. Springer-Verlag, 2001.
25. Tatsuaki Okamoto and David Pointcheval. REACT: Rapid Enhanced-security Asymmetric Cryptosystem Transform. In David Naccache, editor, *CT-RSA 2001*, volume 2020 of *LNCS*, pages 159–175. Springer-Verlag, 2001.
26. Duong Hieu Phan and David Pointcheval. About the security of ciphers (semantic security and pseudo-random permutations). In Helena Handschuh and Anwar Hasan, editors, *SAC 2004*, volume 3357 of *LNCS*, pages 182–197. Springer-Verlag, 2004.
27. Ryuichi Sakai and Masao Kasahara. ID based cryptosystems with pairing on elliptic curve. Cryptology ePrint Archive, Report 2003/054, 2003. `http://eprint.iacr.org/`.
28. Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266. Springer-Verlag, 1997.
29. R. Steinfeld, J. Baek, and Y. Zheng. On the necessity of strong assumptions for the security of a class of asymmetric encryption schemes. In *ACISP 2002*, volume 2384 of *LNCS*, pages 241–256. Springer-Verlag, 2002.