A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack

Ronald Cramer

Institute for Theoretical Computer Science, ETH Zurich, 8092 Zurich, Switzerland

`cramer@inf.ethz.ch`

Victor Shoup

IBM Zurich Research Laboratory, Säumerstr. 4, 8803 Rüschlikon, Switzerland

`sho@zurich.ibm.com`

May 1998

Abstract

A new public key cryptosystem is proposed and analyzed. The scheme is quite practical, and is provably secure against adaptive chosen ciphertext attack under standard intractability assumptions. There appears to be no previous cryptosystem in the literature that enjoys both of these properties simultaneously.

1 Introduction

In this paper, we present and analyze a new public key cryptosystem that is provably secure against adaptive chosen ciphertext attack (as defined by Rackoff and Simon [20]). The scheme is quite practical, requiring just a few exponentiations over a group. Moreover, the proof of security relies only on a standard intractability assumption, namely, the hardness of the Diffie-Hellman decision problem in the underlying group.

The hardness of the Diffie-Hellman decision problem is essentially equivalent to the semantic security of the basic El Gamal encryption scheme [12]. Thus, with just a bit more computation, we get security against adaptive chosen ciphertext attack, whereas the basic El Gamal scheme is completely insecure against adaptive chosen ciphertext attack. Actually, the basic scheme we describe also requires a universal one-way hash function. In a typical implementation, this can be efficiently constructed without extra assumptions; however, we also present a hash-free variant as well.

While there are several provably secure encryption schemes in the literature, they are all quite impractical. Also, there are several practical cryptosystems that have been proposed, but none of them has been proven secure under standard intractability assumptions. The significance of our contribution is that it provides a scheme that is provably secure and practical at the same time. There appears to be no other encryption scheme in the literature that enjoys both of these properties simultaneously.

Chosen Ciphertext Security

Semantic security, defined by Goldwasser and Micali [14], captures the intuition that an adversary should not be able to obtain any partial information about a message given its encryption. However, this guarantee of secrecy is only valid when the adversary is completely passive, i.e., can only eavesdrop. Indeed, semantic security offers no guarantee of secrecy at all if an adversary can mount an active attack, i.e., inject messages into a network or otherwise influence the behavior of parties in the network.

To deal with active attacks, Rackoff and Simon [20] defined the notion of security against an *adaptive chosen ciphertext attack*. If an adversary can inject messages into a network, these messages may be encryptions, and the adversary may be able to extract partial information about the corresponding cleartexts through its interactions with the parties in the network. Rackoff and Simon's definition models this type of attack by simply allowing an adversary to obtain decryptions of its choice, i.e., the adversary has access to a "decryption oracle." Now, given an encryption of a message—the "target" ciphertext—we want to guarantee that the adversary cannot obtain any partial information about the message. To achieve this, we have to restrict the adversary's behavior in some way, otherwise the adversary could simply submit the target ciphertext itself to the decryption oracle. The restriction proposed by Rackoff and Simon is the weakest possible: the adversary is not allowed to submit the target ciphertext itself to the oracle; however, it may submit any other ciphertext, including ciphertexts that are related to the target ciphertext.

A different notion of security against active attacks, called *non-malleability*, was proposed by Dolev, Dwork, and Naor [9]. Here, the adversary also has access to a decryption oracle, but his goal is not to obtain partial information about the target ciphertext, but rather, to create another encryption of a different message that is related in some interesting way to the original, encrypted message. For example, for a non-malleable encryption scheme, given an encryption of n, it should be infeasible to create an encryption of $n + 1$. It turns out that non-malleability and security against adaptive chosen ciphertext attack are equivalent [10].

A cryptosystem secure against adaptive chosen ciphertext attack is a very powerful cryptographic primitive. It is essential in designing protocols that are secure against active adversaries. For example, this primitive is used in protocols for authentication and key exchange [11, 10, 2] and in protocols for escrow, certified e-mail, and more general fair exchange [1, 22]. The practical importance of this primitive is also highlighted by the adoption of Bellare and Rogaway's OAEP scheme [4] (a practical but only heuristically secure scheme) as an internet encryption standard and for use in the SET protocol for electronic commerce.

There are also intermediate notions of security, between semantic security and adaptive chosen ciphertext security. Naor and Yung [19] propose an attack model where the adversary has access to the decryption oracle only *prior* to obtaining the target ciphertext, and the goal of the adversary is to obtain partial information about the encrypted message. Naor and Yung called this type of attack a *chosen ciphertext attack*; it has also been called a "lunch-time" or "midnight" attack. In this paper, we will always use the phrase *adaptive* chosen ciphertext attack for Rackoff and Simon's definition, to distinguish it from Naor and Yung's definition.

Previous Work

Provably Secure Schemes. Naor and Yung [19] presented the first scheme provably secure against lunch-time attacks. Subsequently, Dolev, Dwork, and Naor [9] presented a scheme that is provably secure against adaptive chosen ciphertext attack.

Rackoff and Simon [20] present and prove the security of an encryption scheme, but their scheme is actually not a public key scheme in the traditional sense: in their scheme, *all users*—both senders and receivers—require public keys, and moreover, a trusted center is required to perform certain functions. In contrast, all other schemes mentioned in this paper, including our own, are traditional public key systems: encryption is a probabilistic function of the message and the receiver's public key, decryption is a function of the ciphertext and the receiver's secret key, and no trusted center is required. This distinction can be important: adding extra system requirements as in the Rackoff and Simon scheme can greatly restrict the range of application of the scheme.

All of the previously known schemes provably secure under standard intractability assumptions are completely impractical (albeit polynomial time), as they rely on general and expensive constructions for non-interactive zero-knowledge proofs. This includes non-standard schemes like Rackoff and Simon's as well.

Practical Schemes. Damgard [8] proposed a practical scheme that he conjectured to be secure against lunch-time attacks; however, this scheme is not known to be provably secure, and is in fact demonstrably insecure against adaptive chosen ciphertext attack.

Zheng and Seberry [24] proposed practical schemes that are conjectured to be secure against chosen ciphertext attack, but again, no proof based on standard intractability assumptions is known. Lim and Lee [16] also proposed practical schemes that were later broken by Frankel and Yung [13].

Bellare and Rogaway [3, 4] have presented practical schemes for which they give heuristic proofs of adaptive chosen ciphertext security; namely, they prove security in an idealized model of computation, the so-called *random oracle* model, wherein a hash function is represented by a random oracle.

Shoup and Gennaro [22] also give El Gamal-like schemes that are secure against adaptive chosen ciphertext attack in the random oracle model, and that are also amenable to efficient threshold decryption.

We stress that although a security proof in the random oracle model is of some value, it is still only a heuristic proof. In particular, these types of proofs do not rule out the possibility of breaking the scheme without breaking the underlying intractability assumption. Nor do they even rule out the possibility of breaking the scheme without finding some kind of weakness in the hash function, as recently shown by Canetti, Goldreich, and Halevi [7].

Outline of paper

In §2 we review the basic definitions that we need for security and intractability assumptions. In §3 we outline our basic scheme, and in §4 we prove its security. In §5 we discuss some implementation details and variations on the basic scheme.

2 Definitions

2.1 Security against adaptive chosen ciphertext attack

We recall Rackoff and Simon's definition.

Security is defined via the following game played by the adversary.

First, the encryption scheme's key generation algorithm is run, with a security parameter as input. Next, the adversary makes arbitrary queries to a "decryption oracle," decrypting ciphertexts of his choice.

Next the adversary chooses two messages, m_0, m_1, and sends these to an "encryption oracle." The encryption oracle chooses a bit $b \in \{0,1\}$ at random, and encrypts m_b. The corresponding ciphertext is given to the adversary (the internal coin tosses of the encryption oracle, in particular b, are not in the adversary's view).

After receiving the ciphertext from the encryption oracle, the adversary continues to query the decryption oracle, subject only to the restriction that the query must be different than the output of the encryption oracle.

At the end of the game, the adversary outputs $b' \in \{0,1\}$, which is supposed to be the adversary's guess of the value b. If the probability that $b' = b$ is $1/2 + \epsilon$, then the adversary's *advantage* is defined to be ϵ.

The cryptosystem is said to be secure against adaptive chosen ciphertext attack if the advantage of any polynomial-time adversary is negligible (as a function of the security parameter).

2.2 The Diffie-Hellman Decision Problem

There are several equivalent formulations of the Diffie-Hellman decision problem. The one that we shall use is the following.

Let G be a group of large prime order q, and consider the following two distributions:

- the distribution $\mathbf{R}$ of random quadruples $(g_1, g_2, u_1, u_2) \in G^4$;
- the distribution $\mathbf{D}$ of quadruples $(g_1, g_2, u_1, u_2) \in G^4$, where g_1, g_2 are random, and $u_1 = g_1^r$ and $u_2 = g_2^r$ for random $r \in \mathbf{Z}_q$.

An algorithm that solves the Diffie-Hellman decision problem is a statistical test that can effectively distinguish these two distributions. That is, given a quadruple coming from one of the two distributions, it should output 0 or 1, and there should be a non-negligible difference between (a) the probability that it outputs a 1 given an input from $\mathbf{R}$, and (b) the probability that it outputs a 1 given an input from $\mathbf{D}$. The Diffie-Hellman decision problem is hard if there is no such polynomial-time statistical test.

This formulation of the Diffie-Hellman decision problem is equivalent to several others. First, making the substitution

$$g_1 \to g,\ g_2 \to g^x,\ u_1 \to g^y,\ u_2 \to g^{xy},$$

one sees that this is equivalent to distinguishing Diffie-Hellman triples (g^x, g^y, g^{xy}) from non-Diffie-Hellman triples (g^x, g^y, g^z). Note that by a trivial random self-reducibility property, it does not matter if the base g is random or fixed.

Second, although we have described it as a problem of distinguishing two distributions, the Diffie-Hellman decision problem is equivalent to the worst-case decision problem: given (g^x, g^y, g^z), decide—with negligible error probability—if $z = xy \bmod q$. This equivalence follows immediately from a random self-reducibility property first observed by Stadler [23] and later by Naor and Reingold [17].

Related to the Diffie-Hellman decision problem is the Diffie-Hellman problem (given g, g^x and g^y, compute g^{xy}), and the discrete logarithm problem (given g and g^x, compute x).

There are obvious polynomial-time reductions from the Diffie-Hellman decision problem to the Diffie-Hellman problem, and from the Diffie-Hellman problem to the discrete logarithm problem, but reductions in the reverse direction are not known. Moreover, these reductions are essentially the only known methods of solving the Diffie-Hellman or Diffie-Hellman decision problems. All three problems are widely conjectured to be hard, and have been used as assumptions in proving the security of a variety of cryptographic protocols. Some heuristic evidence for the hardness of all of these problems is provided in [21], where it is shown that they are hard in a certain natural, structured model of computation. See [23, 17, 6] for further applications and discussion of the Diffie-Hellman decision problem.

Note that the hardness of the Diffie-Hellman decision problem is equivalent to the semantic security of the basic El Gamal encryption scheme. Recall that in the basic El Gamal scheme, we encrypt a message $m \in G$ as $(g^r, h^r m)$, where h is the public key of the recipient.

On the one hand, if the Diffie-Hellman decision problem is hard, then the group element h^r could be replaced by a random group element without changing significantly the behavior of the attacker; however, if we perform this substitution, the message m is perfectly hidden, which implies security.

On the other hand, if the Diffie-Hellman decision problem can be efficiently solved, then an attacker can break El Gamal as follows. The attacker chooses two messages m_0, m_1, giving these to an encryption oracle. The encryption oracle produces an encryption $(u, e) = (g^r, h^r m_b)$, where $b \in \{0, 1\}$ is chosen at random. The attacker's task is to determine b, which he can do by simply determining which of $(u, h, e/m_0)$ and $(u, h, e/m_1)$ is a Diffie-Hellman triple.

Note that the basic El Gamal scheme is completely insecure against adaptive chosen ciphertext attack. Indeed, given an encryption (u, e) of a message m, we can feed the $(u, g \cdot e)$ to the decryption oracle, which gives us $g \cdot m$.

2.3 Collision-resistant Hash Functions

A family of hash functions is said to be *collision resistant* if upon drawing a function H at random from the family, it is infeasible for an adversary to find two *different* inputs x and y such that $H(x) = H(y)$.

A weaker notion is that of a *universal one-way* family of hash functions [18]. Here, it should be infeasible for an adversary to choose an input x, draw a random hash function H, and then find a different input y such that $H(x) = H(y)$. Such hash function families are also called *target collision* *resistant.* See [5] for recent results and further discussion.

3 The Basic Scheme

We assume that we have a group G of prime order q, where q is large. We also assume that cleartext messages are (or can be encoded as) elements of G (although this condition can be relaxed—see §5.2). We also use a universal one-way family of hash functions that map long bit strings to elements of $\mathbf{Z}_q$ (although we can do without this—see §5.3).

Key Generation. The key generation algorithm runs as follows. Random elements $g_1, g_2 \in G$ are chosen, and random elements

$$x_1, x_2, y_1, y_2, z \in \mathbf{Z}_q$$

are also chosen. Next, the group elements

$$c = g_1^{x_1} g_2^{x_2},\ d = g_1^{y_1} g_2^{y_2},\ h = g_1^{z}$$

are computed. Next, a hash function H is chosen from the family of universal one-way hash functions. The public key is (g_1, g_2, c, d, h, H), and the private key is (x_1, x_2, y_1, y_2, z).

Encryption. Given a message $m \in G$, the encryption algorithm runs as follows. First, it chooses $r \in \mathbf{Z}_q$ at random. Then it computes

$$u_1 = g_1^r,\ u_2 = g_2^r,\ e = h^r m,\ \alpha = H(u_1, u_2, e),\ v = c^r d^{r\alpha}.$$

The ciphertext is

$$(u_1, u_2, e, v).$$

Decryption. Given a ciphertext (u_1, u_2, e, v), the decryption algorithm runs as follows. It first computes $\alpha = H(u_1, u_2, e)$, and tests if

$$u_1^{x_1+y_1\alpha} u_2^{x_2+y_2\alpha} = v.$$

If this condition does not hold, the decryption algorithm outputs "reject"; otherwise, it outputs

$$m = e/u_1^z.$$

We first verify that this is an encryption scheme, in the sense that the decryption of an encryption of a message yields the message. Since $u_1 = g_1^r$ and $u_2 = g_2^r$, we have

$$u_1^{x_1} u_2^{x_2} = g_1^{rx_1} g_2^{rx_2} = c^r.$$

Likewise, $u_1^{y_1} u_2^{y_2} = d^r$ and $u_1^z = h^r$ Therefore, the test performed by the decryption algorithm will pass, and the output will be $e/h^r = m$.

4 Proof of Security

In this section, we prove the following theorem.

Theorem 1 *The above cryptosystem is secure against adaptive chosen ciphertext attack assuming that (1) the hash function H is chosen from a universal one-way family, and (2) the Diffie-Hellman decision problem is hard in the group G.*

To prove the theorem, we will assume that there is an adversary that can break the cryptosystem, and that the hash family is universal one-way, and show how to use this adversary to construct a statistical test for the Diffie-Hellman decision problem.

For the statistical test, we are given (g_1, g_2, u_1, u_2) coming from either the distribution **R** or **D**. At a high level, our construction works as follows. We build a simulator that simulates the joint distribution consisting of adversary's view in its attack on the cryptosystem, and the hidden bit b generated by the generated oracle (which is not a part of the adversary's view).

We will show that if the input comes from **D**, the simulation will be nearly perfect, and so the adversary will have a non-negligible advantage in guessing the hidden bit b. We will also show that if the input comes from **R**, then the adversary's view is essentially independent of b, and therefore the adversary's advantage is negligible. This immediately implies a statistical test distinguishing **R** from **D**: run the simulator and adversary together, and if the simulator outputs b and the adversary outputs b', the distinguisher outputs 1 if $b = b'$, and 0 otherwise.

We now give the details of the simulator. The input to the simulator is (g_1, g_2, u_1, u_2). The simulator runs the following key generation algorithm, using the given g_1, g_2. The simulator chooses

$$x_1, x_2, y_1, y_2, z_1, z_2 \in \mathbf{Z}_q$$

at random, and computes

$$c = g_1^{x_1} g_2^{x_2},\ d = g_1^{y_1} g_2^{y_2},\ h = g_1^{z_1} g_2^{z_2}.$$

The simulator also chooses a hash function H at random. The public key that the adversary sees is (g_1, g_2, c, d, h, H). The simulator knows $(x_1, x_2, y_1, y_2, z_1, z_2)$.

Note that the simulator's key generation algorithm is slightly different from the key generation algorithm of the actual cryptosystem; in the latter, we essentially fix $z_2 = 0$.

The simulator answers decryption queries as in the actual attack, except that it computes $m = e/(u_1^{z_1} u_2^{z_2})$.

We now describe the simulation of the encryption oracle. Given m_0, m_1, the simulator chooses $b \in \{0, 1\}$ at random, and computes

$$e = u_1^{z_1} u_2^{z_2} m_b,\ \alpha = H(u_1, u_2, e),\ v = u_1^{x_1 + y_1 \alpha} u_2^{x_2 + y_2 \alpha},$$

and outputs

$$(u_1, u_2, e, v).$$

That completes the description of the simulator. As we will see, when the input to the simulator comes from $\mathbf{D}$, the output of the encryption oracle is a perfectly legitimate ciphertext; however, when the input to the simulator comes from $\mathbf{R}$, the output of the decryption oracle will not be legitimate, in the sense that $\log_{g_1} u_1 \neq \log_{g_2} u_2$. This is not a problem, and indeed, it is crucial to the proof of security.

The theorem now follows immediately from the following two lemmas.

Lemma 1 *When the simulator's input comes from* $\mathbf{D}$*, the joint distribution of the adversary's view and the hidden bit* b *is is statistically indistinguishable from that in the actual attack.*

Consider the joint distribution of the adversary's view and the bit b when the input comes from the distribution $\mathbf{D}$. Say $u_1 = g_1^r$ and $u_2 = g_2^r$.

It is clear in this case that the output of the encryption oracle has the right distribution, since $u_1^{x_1} u_2^{x_2} = c^r$, $u_1^{y_1} u_2^{y_2} = d^r$, and $u_1^{z_1} u_2^{z_2} = h^r$; indeed, these equations imply that $e = m_b h^r$ and $v = c^r d^{r\alpha}$, and α itself is already of the right form.

To complete the proof, we need to argue that the output of the decryption oracle has the right distribution. Let us call $(u_1', u_2', e', v') \in G^4$ a *valid ciphertext* if $\log_{g_1} u_1' = \log_{g_2} u_2'$.

Note that if a ciphertext is valid, with $u_1' = g_1^{r'}$ and $u_2' = g_2^{r'}$, then $h^{r'} = (u_1')^{z_1}(u_2')^{z_2}$; therefore, the decryption oracle outputs $e/h^{r'}$, just as it should. Consequently, the lemma follows immediately from the following:

Claim. The decryption oracle—in both an actual attack against the cryptosystem and in an attack against the simulator—rejects all invalid ciphertexts, except with negligible probability.

We now prove this claim by considering the distribution of the point $\mathbf{P} = (x_1, x_2, y_1, y_2) \in \mathbf{Z}_q^4$, conditioned on the adversary's view. Let $\log(\cdot)$ denote $\log_{g_1}(\cdot)$, and let $w = \log g_2$.

From the adversary's view, $\mathbf{P}$ is a random point on the plane $\mathcal{P}$ formed by intersecting the hyperplanes

$$\log c = x_1 + wx_2 \tag{1}$$

and

$$\log d = y_1 + wy_2. \tag{2}$$

These two equations come from the public key. The output from the encryption oracle does not constrain $\mathbf{P}$ any further, as the hyperplane defined by

$$\log v = rx_1 + wrx_2 + \alpha ry_1 + \alpha rwy_2 \tag{3}$$

contains $\mathcal{P}$.

Now suppose the adversary submits an invalid ciphertext (u_1', u_2', v', e') to the decryption oracle, where $\log u_1' = r_1'$ and $\log u_2' = wr_2'$, with $r_1' \neq r_2'$. The decryption oracle will reject, unless $\mathbf{P}$ happens to lie on the hyperplane $\mathcal{H}$ defined by

$$\log v' = r_1'x_1 + wr_2'x_2 + \alpha' r_1' y_1 + \alpha' r_2' wy_2, \tag{4}$$

where $\alpha' = H(u_1', u_2', e')$. But it is clear that the equations (1), (2), and (4) are linearly independent, and so $\mathcal{H}$ intersects the plane $\mathcal{P}$ at a line.

It follows that the first time the adversary submits an invalid ciphertext, the decryption oracle rejects with probability $1 - 1/q$. This rejection actually constrains the point $\mathbf{P}$, puncturing the plane $\mathcal{H}$ at a line. Therefore, for $i = 1, 2, \ldots$, the ith invalid ciphertext submitted by the adversary will be rejected with probability at least $1 - 1/(q - i + 1)$. From this it follows that the decryption oracle rejects all invalid ciphertexts, except with negligible probability.

Lemma 2 *When the simulator's input comes from* $\mathbf{R}$*, the distribution of the hidden bit* b *is (essentially) independent from the adversary's view.*

Let $u_1 = g_1^{r_1}$ and $u_2 = g_1^{wr_2}$. We may assume that $r_1 \neq r_2$, since this occurs except with negligible probability. The lemma follows immediately from the following two claims.

Claim 1. If the decryption oracle rejects all invalid ciphertexts during the attack, then the distribution of the hidden bit b *is independent of the adversary's view.*

To see this, consider the point $\mathbf{Q} = (z_1, z_2) \in \mathbf{Z}_q^2$. At the beginning of the attack, this is a random point on the line

$$\log h = z_1 + wz_2, \tag{5}$$

determined by the public key. Moreover, if the decryption oracle only decrypts valid ciphertexts (u_1', u_2', e', v'), then the adversary obtains only linearly dependent relations $r' \log h = r'z_1 + r'wz_2$ (since $(u_1')^{z_1}(u_2')^{z_2} = g_1^{r'z_1} g_2^{r'z_2} = h^{r'}$). Thus, no further information about $\mathbf{Q}$ is leaked.

Consider now the output (u_1, u_2, e, v) of the simulator's encryption oracle. We have $e = \epsilon \cdot m_b$, where $\epsilon = u_1^{z_1} u_2^{z_2}$. Now, consider the equation

$$\log \epsilon = r_1 z_1 + wr_2 z_2. \tag{6}$$

Clearly, (5) and (6) are linearly independent, and so the conditional distribution of ϵ—conditioning on b and everything in the adversary's view other than e—is uniform. In other words, ϵ is a perfect one-time pad. It follows that b is independent of the adversary's view.

Claim 2. The decryption oracle will reject all invalid ciphertexts, except with negligible probability.

As in the proof of Lemma 1, we study the distribution of $\mathbf{P} = (x_1, x_2, y_1, y_2) \in \mathbf{Z}_q^4$, conditioned on the adversary's view. From the adversary's view, this is a random point on the line $\mathcal{L}$ formed by intersecting the hyperplanes (1), (2), and

$$\log v = r_1 x_1 + w r_2 x_2 + \alpha r_1 y_1 + \alpha w r_2 y_2. \tag{7}$$

Equation (7) comes from the output of the encryption oracle.

Now assume that the adversary submits an invalid ciphertext $(u_1', u_2', e', v') \neq (u_1, u_2, e, v)$, where $\log u_1' = r_1'$ and $\log u_2' = w r_2'$, with $r_1' \neq r_2'$. Let $\alpha' = H(u_1', u_2', e')$.

There are three cases we consider.

Case 1. $(u_1', u_2', e') = (u_1, u_2, e)$. In this case, the hash values are the same, but $v' \neq v$ implies that the decryption oracle will certainly reject.

Case 2. $(u_1', u_2', e') \neq (u_1, u_2, e)$ and $\alpha' \neq \alpha$.

The decryption oracle will reject unless the point $\mathbf{P}$ lies on the hyperplane $\mathcal{H}$ defined by (4). However, the equations (1), (2), (7), and (4) are linearly independent. This can be verified by observing that

$$\det \begin{pmatrix} 1 & w & 0 & 0 \\ 0 & 0 & 1 & w \\ r_1 & w r_2 & \alpha r_1 & \alpha w r_2 \\ r_1' & w r_2' & \alpha' r_1' & \alpha' w r_2' \end{pmatrix} = w^2 (r_2 - r_1)(r_2' - r_1')(\alpha - \alpha') \neq 0.$$

Thus, $\mathcal{H}$ intersects the line $\mathcal{L}$ at a point, from which it follows (as in the proof of Lemma 1) that the decryption oracle rejects, except with negligible probability.

Case 3. $(u_1', u_2', e') \neq (u_1, u_2, e)$ and $\alpha' = \alpha$. We argue that if this happens with nonnegligible probability, then in fact, the family of hash functions is not universal one-way—a contradiction. Note that if we made the stronger assumption of collision resistance, there would be essentially nothing to prove, but with the weaker universal one-way assumption, an argument is needed. We use the adversary to break the universal one-way hash function as follows. We modify the encryption oracle in the simulator, so that it outputs (u_1, u_2, e, v) as before, except that now, $e \in G$ is simply chosen completely at random. Up until such time that a collision occurs, the adversary's view in this modified simulation is statistically indistinguishable from the view in the original simulation, and so the adversary will also find a collision with nonnegligible probability in the modified simulation. But the argument (u_1, u_2, e) to H is independent of H, and in particular, we can choose it *before* choosing H.

5 Implementation Details and Variations

In this section, we briefly discuss some implementation details and possible variations of the basic encryption scheme.

5.1 A simple implementation

We choose a large prime p such that $p - 1 = 2q$, where q is also prime. The group G is the subgroup of order q in $\mathbf{Z}_p^*$. We restrict a message to be an element of the set $\{1, \ldots, q\}$, and "encode" it by squaring it modulo p, giving us an element in G. We can recover a message from its encoding by computing the unique square root of its encoding modulo p that is in the set $\{1, \ldots, q\}$.

For the hash function, one could use a function like SHA-1, or possibly some keyed variant, and make the appropriate collision-resistance assumption. However, it is only marginally more expensive to do the following, which is based only on the hardness of discrete logarithms in G. Say we want to hash a bit string to an integer mod q. Write the bit string as a sequence $(a_1, \ldots, a_k)$, with each $a_i \in \{0, \ldots, q-1\}$. To define the hash function, choose $h_1, \ldots, h_k$ in G at random. The hash of $(a_1, \ldots, a_k)$ is then the least non-negative residue of $\pm h_1^{a_1} \cdots h_k^{a_k} \in \mathbf{Z}_p^*$, where the sign is chosen so that this value is in $\{1, \ldots, q\}$.

This hash function is collision resistant, provided computing discrete logarithms in G is hard. To see this, note that from a collision, we obtain a nonzero sequence $(a_1, \ldots, a_k)$ mod q such that

$$h_1^{a_1} \cdots h_k^{a_k} \in \{1, -1\} \cap G = \{1\}.$$

Using a standard argument, it is easy to see that finding such a relation is equivalent to computing discrete logarithms.

Note that the group elements g_1, g_2 and $h_1, \ldots, h_k$ can be system-wide parameters, used by all users of the system.

5.2 A hybrid implementation

It would be more practical to work in a smaller subgroup, and it would be nice to have a more flexible and efficient way to encode messages.

To do this, assume we have a symmetric-key cipher C with a key length of l bits. Now choose a large prime p such that $p - 1 = qm$, where q is a $3l$-bit prime. The group G is the subgroup of order q in $\mathbf{Z}_p^*$. A message in this scheme is just an arbitrary bit string. To encrypt a message m, we modify our encryption algorithm, computing $e = C_K(m)$, where the encryption key K is computed by hashing h^r to an l-bit string with a public 2-universal hash function.

For the hash function H used in the encryption scheme, something like SHA-1, possibly keyed, would be appropriate.

The security of this variant is easily proved using the techniques of this paper, along with the left-over hash lemma [15], assuming the cipher C is semantically secure.

5.3 A hash-free variant

We can actually eliminate the hash function H from the scheme, so that the security can be based strictly on the Diffie-Hellman decision problem for an arbitrary group G. Suppose the strings we need to hash in the original scheme are of the form $(a_1, \ldots, a_k)$, where $0 \leq a_i < p$. In the modified scheme, we replace the group element d in the public key by $d_1, \ldots, d_k$. For $1 \leq i \leq k$, we have $d_i = g_1^{y_{i1}} g_2^{y_{i2}}$, where y_{i1} and y_{i2} are random elements of $\mathbf{Z}_q$ included in the secret key. When encrypting, we compute

$$v = c^r \prod_{i=1}^{k} d_i^{a_i r},$$

and when decrypting, we verify that

$$v = u_1^{x_1 + \sum_{i=1}^{k} a_i y_{i1}} u_2^{x_2 + \sum_{i=1}^{k} a_i y_{i2}}.$$

Using the same proof techniques as for the basic scheme, it is straightforward to prove that this modified version is secure against adaptive chosen ciphertext attack, assuming the Diffie-Hellman decision problem in G is hard.

5.4 A "lite" version secure against lunch-time attacks

To achieve security against lunch-time attacks only, one can simplify the basic scheme significantly, essentially by eliminating d, y_1, y_2, and the hash function H. When encrypting, we compute $v = c^r$, and when decrypting, we verify that $v = u_1^{x_1} u_2^{x_2}$.

Acknowledgments

We would like to thank Moni Naor for his very useful comments on an earlier draft of this paper, and in particular, for pointing out that a universal one-way hash function is sufficient to prove the security of our basic scheme, and for suggesting the hash-free variant in §5.3.

References

[1] N. Asokan, V. Shoup, and M. Waidner. Optimistic fair exchange of digital signatures. In *Advances in Cryptology–Eurocrypt '98*, 1998.

[2] M. Bellare, R. Canetti, and H. Krawczyk. A modular approach to the design and analysis of authentication and key exchange protocols. In *30th Annual ACM Symposium on Theory of Computing*, 1998.

[3] M. Bellare and P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *First ACM Conference on Computer and Communications Security*, pages 62–73, 1993.

[4] M. Bellare and P. Rogaway. Optimal asymmetric encryption. In *Advances in Cryptology—Crypto '94*, pages 92–111, 1994.

[5] M. Bellare and P. Rogaway. Collision-resistant hashing: towards making UOWHFs practical. In *Advances in Cryptology–Crypto '97*, 1997.

[6] D. Boneh and R. Venkatesan. Hardness of computing the most significant bits of secret keys in Diffie-Hellman and related schemes. In *Advances in Cryptology–Crypto '96*, pages 129–142, 1996.

[7] R. Canetti, O. Goldreich, and S. Halevi. The random oracle model, revisted. In *30th Annual ACM Symposium on Theory of Computing*, 1998. To appear.

[8] I. Damgard. Towards practical public key cryptosystems secure against chosen ciphertext attacks. In *Advances in Cryptology–Crypto '91*, pages 445–456, 1991.

[9] D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. In *23rd Annual ACM Symposium on Theory of Computing*, pages 542–552, 1991.

[10] D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography, 1998. Manuscript (updated, full length version of STOC paper).

[11] C. Dwork and M. Naor. Method for message authentication from non-malleable cryptosystems, 1996. U. S. Patent No. 05539826.

[12] T. El Gamal. A public key cryptosystem and signature scheme based on discrete logarithms. *IEEE Trans. Inform. Theory*, 31:469–472, 1985.

[13] Y. Frankel and M. Yung. Cryptanalysis of immunized LL public key systems. In *Advances in Cryptology–Crypto '95*, pages 287–296, 1995.

[14] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.

[15] R. Impagliazzo, L. Levin, and M. Luby. Pseudo-random number generation from any one-way function. In *21st Annual ACM Symposium on Theory of Computing*, pages 12–24, 1989.

[16] C. H. Lim and P. J. Lee. Another method for attaining security against adaptively chosen ciphertext attacks. In *Advances in Cryptology–Crypto '93*, pages 420–434, 1993.

[17] M. Naor and O. Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *38th Annual Symposium on Foundations of Computer Science*, 1997.

[18] M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *21st Annual ACM Symposium on Theory of Computing*, 1989.

[19] M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd Annual ACM Symposium on Theory of Computing*, pages 427–437, 1990.

[20] C. Rackoff and D. Simon. Noninteractive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Advances in Cryptology–Crypto '91*, pages 433–444, 1991.

[21] V. Shoup. Lower bounds for discrete logarithms and related problems. In *Advances in Cryptology–Eurocrypt '97*, 1997.

[22] V. Shoup and R. Gennaro. Securing threshold cryptosystems against chosen ciphertext attack. In *Advances in Cryptology–Eurocrypt '98*, 1998.

[23] M. Stadler. Publicly verifiable secrete sharing. In *Advances in Cryptology–Eurocrypt '96*, pages 190–199, 1996.

[24] Y. Zheng and J. Seberry. Practical approaches to attaining security against adaptively chosen ciphertext attacks. In *Advances in Cryptology–Crypto '92*, pages 292–304, 1992.