On the Security of a Williams Based Public Key Encryption Scheme

Siguna Müller*

University of Klagenfurt, Dept. of Math., A-9020 Klagenfurt, Austria
siguna.mueller@uni-klu.ac.at

Abstract. In 1984, H.C. Williams introduced a public key cryptosystem whose security is as intractable as factorization. Motivated by some strong and interesting cryptographic properties of the intrinsic structure of this scheme, we present a practical modification thereof that has very strong security properties. We establish, and prove, a generalization of the "sole-samplability" paradigm of Zheng-Seberry (1993) which is reminiscent of the plaintext-awareness concept of Bellare et. al. The assumptions that we make are both well-defined and reasonable. In particular, we do not model the functions as random oracles. In essence, the proof of security is based on the factorization problem of any large integer $n = pq$ and Canetti's "oracle hashing" construction introduced in 1997. Another advantage of our system is that we do not rely on any special structure of the modulus $n = pq$, nor do we require any specific form of the primes p and q. As our main result we establish a model which implies security attributes even stronger than semantic security against chosen ciphertext attacks.

Keywords: Chosen Ciphertext Security, Plaintext Awareness, (Weak)-Sole-Samplability, Factorization Intractability, Oracle Hashing, Williams' Encryption Scheme

1 Introduction and Summary

1.1 Provable Security and Attack Models

A desirable property of any cryptosystem is a proof that breaking it is as difficult as solving a computational problem that is widely believed to be difficult. A cryptographic scheme is provably secure if an attack on the scheme implies an attack on the underlying primitives it employs. While RSA is undoubtedly the most well-known and widely used public-key cryptosystem, it is not known if breaking RSA is as difficult as factoring (cf. [6]). A variety of factorization equivalent RSA modifications have been proposed which are essentially based on the same idea of unambiguous decryption (cf. also [18]). The sender can manipulate the decoder to decrypt a 'wrong message' which then can be used to factorize the modulus. Because of this problem, all these systems are vulnerable

* Supported by the Austrian Science Fund (FWF), P 13088-MAT and P 14472-MAT

K. Kim (Ed.): PKC 2001, LNCS 1992, pp. 1–18, 2001.

to a *chosen ciphertext attack* (CCA). Under such an attack the adversary selects the ciphertext and is then given the corresponding plaintext. The strongest such attack is known as the *adaptive* CCA [19], in which an attacker can access a decryption oracle on arbitrary ciphertexts (except for the target ciphertexts which he is challenged with).

It is known that plain RSA can broken under a CCA [21], which allows total recovery of a complete plaintext, resp. generation of a complete signature on an entire message. But RSA is also vulnerable to attacks that compromise the semantic security of the scheme. An adaptive CCA can successfully be mounted on some randomized versions of RSA (PKC # 1), when only partial information of the plaintext is leaked [5].

The underlying goal of any encryption scheme is to achieve *semantic security* (informally, 'whatever can be computed by an attacker about the plaintext given an object ciphertext, can also be computed without the object ciphertext') under strong attack models (such as CCA) under well-specified assumptions and primitives.

1.2 The General Goal of this Paper

The two most often applied cryptographic primitives are the Diffie-Hellman (DH) problem and the factorization problem. There are a number of systems secure against CCA which are based on the DH problems, e.g., on the decisional DH and the existence of a collision resistant hash function [10], on the decisional DH in the random oracle model (ROM) [22], and on the computational DH in the ROM [1,17]. Also, suggestions have been made which are based on various new primitives (cf. e.g. [17]), but no encryption scheme secure against CCA has been published yet which utilizes the factorization utility of arbitrary numbers in a model without random oracles. Very recently, proposals have been made [16,17] of encryption schemes whose security rely on the ROM, and additionally require the very specific structure of the modulus, $n = p^2q$.

Most of the above methods require random oracles. Although the ROM is a convenient setting, we do not have a general mechanism for transforming protocols that are secure in the ROM into protocols that are secure in real life. Actually, it is proved [8] that there are schemes which are secure in the ROM, but have no secure implementation in the "real world". Moreover, we do not even know how to specify the properties for a transformation from the ROM into the real world. A natural goal thus is to design a chosen ciphertext secure system which is practical and *proven secure under well defined intractability assumptions*.

On the other hand, although it is not know to what extend there exist algorithms that can exploit a special structure of the modulus for more efficiently factoring n, it would be desirable to establish a scheme based *on the general factorization primitive* $n = pq$ with p and q arbitrarily. The only factorization equivalent RSA modification known that does not require a specific form of the modulus, nor any special structure of the primes, is the Williams scheme [23].

Our proposed suggestion will consist of enhancing this scheme in order to obtain very strong cryptographic properties.

Decrypting the Williams system is provably equivalent in difficulty to factoring $n = pq$. However, it is vulnerable to a CCA. The main result of the paper is to enhance this system. We will establish some new model which yields properties even stronger than security against CCA.

1.3 Previous Methods for Proving CCA Security

There are several methods for proving security against adaptive CCA [1].

Typically, in the ROM, semantic security against CCA is achieved by proving semantic security against chosen plaintext attacks (CPA) [2], and successively proving that the system is *plaintext aware* [2]. In the first definition given in [3] this basically meant that an adversary cannot produce a ciphertext without knowing ('being able to compute') the corresponding plaintext.

The original definition required some modification. This is due to the way as how a valid ciphertext [2] is created. If its creation involves some internal RO-hash queries, the adversary that produced the ciphertext would not be able to compute the underlying plaintext [2]. The refined definition given in [2] involves some *plaintext extractor* which serves as a *simulator* of the decryption oracle. The extractor is required to find the underlying plaintext to a ciphertext without making any queries to the decryption oracle. A necessary requirement for the plaintext extractor to be successful is that the generation of the ciphertext only involved direct RO-queries. In that case, decryption can be simulated by the extractor, otherwise, it cannot. The fact that there exist some valid ciphertexts that cannot be decrypted by the simulator immediately leads to a smaller success rate of any CCA-attacker and to some loss of 'advantage' [2].

For practical realisations [1,17] the problem firstly consists in showing that such a plaintext extractor exists. Secondly, several probability estimates are necessary to ensure that the *failure probability of the simulator* remains small enough.

Moreover, plaintext awareness (PA), as defined by Bellare et. al. has only been defined in the ROM. In [2] it is argued, why this concept would not make sense in the standard model.

A more direct approach for proving security against CCA was done in [10]. It is shown that if their scheme could be broken under a CCA then this would lead to some method for breaking the underlying primitive (the decisional DH problem).

In [20] it was recently shown that under certain settings security against adaptive CCA is not even enough. Schnorr-Jakobsson demonstrate new and reasonable attack models ('one-more attack') which cannot be covered by CCA

[1] We do not consider the multi user setting, as this would require additional features going beyond the scope of this paper

[2] a valid ciphertext is usually understood as one that passes the validity test and hence does not get rejected by the decryption oracle

security. Indeed, they show that the most important and general attack models can be captured by some sort of proof of knowledge, which is also called *plaintext awareness* in [20] but is different from the definition of PA given in [2]. The PA in [20] also requires that any party that creates a valid ciphertext, must 'know' the secret parameters involved in its creation (for details we refer to [20]). Although the arguments of [20] clearly demonstrate that security against CCA is not sufficient, their method requires the ROM as well and thus cannot be applied to our proposed scheme.

The idea of incorporating some proof of knowledge in proving security against CCA goes back to [11,15]. Although these suggestions do not require the ROM, they are quite impractical as they rely on general and expensive constructions which make these cryptosystems difficult to realize in practice.

The first practical approach for establishing security against adaptive CCA without the ROM was proposed by Zheng-Seberry already in 1993 [26]. They require their encoding functions f to be ***sole-samplable***. Basically, this property means that there is no other way to generate any valid ciphertext than to first choose a plaintext x and evaluate f at x. Thus, an adversary cannot generate a new valid ciphertext without starting from a known plaintext.

The underlying idea is obvious. If the party that generates a valid ciphertext must know the corresponding plaintext then it cannot abuse the system as it must have known the result of any decryption-query to begin with. Sole-samplability is one of the strongest notions of security that exists. It would automatically imply security against non-malleability [11], against adaptive CCA, and also against the one more attack. Additionally, it does not require the ROM.

The problem with the Zheng-Seberry suggestion is that they were not able to prove that their functions are indeed sole-samplable. They merely base the proof of CCA security on this *assumption*. Although their concept seems to be the strongest the difficulty is to actually achieve it.

1.4 The New Method and Our Main Results

We suggest that the underlying primitive to be chosen in the standard model must be some form of sole-samplability. Obviously the most natural and important concept to be established is some 'proof of knowledge', as plaintext awareness in the ROM, or the Schnorr-Jakobsson plaintext awareness in the generic model. Our main results are the following.

- We introduce a comparable notion to the sole-samplability paradigm of Zheng-Seberry. Although the proposed concept is slightly weaker than theirs, it has the advantage that all the established claims can rigorously be proved. We call an encryption scheme ***weak-sole-samplable*** if the following conditions hold. *If C is a valid ciphertext then it either has to be the result of an encryption query, or it has to be the result of some specific function (algorithm) F. In the latter case, this F must be explicitly known, and additionally, it must be possible to efficiently generate the underlying plaintext with publicly available information only.*

- This means, that if there do exist ways to find a valid ciphertext other than running the encryption oracle, then *all these other ways must be explicitly known.* Also, whenever a valid ciphertext was generated by such an (explicitly known) alternative method, then it must be possible to *find the corresponding plaintext, without having to make any decryption-oracle queries.*
- The advantages of this concept are obvious. If all the ways of establishing valid ciphertexts are known, and if none of these cases is possible apart from knowing the underlying plaintext, then the behaviour of any adversary is the same as in the Zheng-Seberry model, which implies extremely strong security attributes. The adversary cannot obtain more information via any decryption-oracle queries, as he must have known the answers to begin with.
- Another advantage is that *this eliminates the need for a simulator and* additionally, the necessity for establishing *the failure probability* of any plaintext extractor (simulator). Any adversary that creates a valid ciphertext, is always successful in finding the corresponding plaintext. There are not any valid ciphertext that can be created without the plaintext.
- We establish the prove of 'weak-sole-samplability' on well-formulated and explicit properties. We *only* require *Canetti's oracle hash functions* and the *general factorization primitive.* To the best of our knowledge this is the first proposal that does not require any structure of the modulus, nor any special form of the primes. Additionally, the scheme remains quite practical and can efficiently be realized by means of very rapid methods for the evaluation of combined Lucas sequences [24,25].

From the RSA family, the only factorization equivalent scheme for arbitrary p, q in the modulus $n = pq$ is the Williams scheme. This system has a number of very interesting properties. Indeed, it was a better understanding of the intrinsic structure of this scheme that lead us to establish the new model and the enhanced security properties. Since 'weak-sole-samplability' is strongly based on underlying properties of the Williams system, we present it in terms of this particular system.

Outline: After a short description of the Williams scheme and some essential properties thereof (section 2), we present the proposed enhanced version (section 3). Semantic security against CPA will be derived in section 4.1. In section 4.2 we finally prove the property 'weak-sole-samplability'.

2 Some Preliminaries

2.1 The Underlying Williams Scheme

Let $\alpha, \overline{\alpha}$ be the distinct roots of $x^2 - Px + Q$ for $P, Q \in \mathbb{Z}$ with $Q \neq 0$ and discriminant $D = P^2 - 4Q$. Then the Lucas sequences of the first and second kind of degree k, are defined by $U_k(P,Q) = \frac{\alpha^k - \overline{\alpha}^k}{\alpha - \overline{\alpha}}$ and $V_k(P,Q) = \alpha^k + \overline{\alpha}^k$, respectively. It follows that these are sequences of integers that fulfill a number of interesting identities and arithmetical properties [25].

Williams [23] utilizes the Lucas sequences for the special case where $P = 2a$, and $Q = 1$. Then, if p is an odd prime, one obtains the fundamental congruence $\alpha^{p-(D/p)} \equiv \overline{\alpha}^{p-(D/p)} \equiv 1 \bmod p$. Analogously as in Rabin's case, the basis of the system is the congruence $\alpha^{(p-(D/p))/2} \equiv \overline{\alpha}^{(p-(D/p))/2} \equiv \pm 1 \bmod p$. Williams develops a method to specify the correct signs. When working modulo $n = pq$, and for e, d the public and private key, respectively, he obtains $\alpha^{2ed} \equiv \overline{\alpha}^{2ed} \equiv \pm\alpha \bmod n$. This then establishes the equivalence between decryption and factoring [23].

Let $n = pq$, where p and q are two large primes. Further, let $s, c \in \mathbb{Z}_n^*$ be chosen such such that $p \equiv -\left(\frac{c}{p}\right) \bmod 4$ and $q \equiv -\left(\frac{c}{q}\right) \bmod 4$, $\gcd(s^2-c, n) = 1$ and $\left(\frac{s^2-c}{n}\right) = -1$. In the following $w \in \mathbb{Z}_n$ is assigned the role of the message to be encrypted.

Let the public encryption key e and the secret decryption key d with $\gcd(e, (p+1)(q+1)) = 1$ be determined according to $ed \equiv \frac{m+1}{2} \bmod m$, where $m = \frac{(p-\left(\frac{c}{p}\right))(q-\left(\frac{c}{q}\right))}{4}$. The numbers n, e, c, s constitute the *Public Key*, whereas the numbers p, q, m, d are kept *secret.* Throughout the paper, let $b_1 = 1$, if $\left(\frac{w^2-c}{n}\right) = 1$, and $b_1 = -1$, if $\left(\frac{w^2-c}{n}\right) = -1$.

Suppose $\gcd(w^2 - c, n) = 1$ and denote $a \equiv a(w), b \equiv b(w) \bmod n$, and $\alpha \equiv \alpha(w) \equiv a + b\sqrt{c} \bmod n$, where [3]

$$\begin{cases} \text{for } b_1 = 1: & a \equiv \frac{w^2+c}{w^2-c}, \quad b \equiv \frac{2w}{w^2-c} \bmod n, \\ \text{for } b_1 = -1: & a \equiv \frac{(w^2+c)(s^2+c)+4csw}{(w^2-c)(s^2-c)}, \; b \equiv \frac{2s(w^2+c)+2w(s^2+c)}{(w^2-c)(s^2-c)} \bmod n. \end{cases} \tag{1}$$

Define the sequences $X_i(a) = \frac{\alpha^i+\overline{\alpha}^i}{2} = \frac{V_i(2a,1)}{2}$, and $Y_i(a,b) = b\frac{\alpha^i-\overline{\alpha}^i}{\alpha-\overline{\alpha}} = bU_i(2a,1)$.

In order to minimize problems concerning the existence of the above multiplicative inverses mod n (cf. [14]) it is preferable to work with a slightly modified version of the original scheme. In the following we will exclusively be applying this modification which essentially consists of reversing numerator and denominator of the original encryption function $X_e(a)/Y_e(a,b) \bmod n$ of [23] and adapting the decryption scheme.

Williams' Encryption: The first step of the encryption process consists of calculating $a(w), b(w)$ from the message w by means of (1). Then w is encoded as

$$E(w) \equiv \frac{Y_e(a(w), b(w))}{X_e(a(w))} \bmod n.$$

The cryptogram C to be transmitted is the triple $[E(w), b_1, b_2]$, where b_1 is defined via $\left(\frac{w^2-c}{n}\right)$ as above and $b_2 \equiv a(w) \bmod 2$, $b_2 \in \{0, 1\}$.

[3] It turns out that the quantities $a(w)$ and $b(w)$ for both cases $b_1 = 1$ and -1 can be comprised into a more comprehensive formula. It can easily be shown that for a and b as above, $a = a(w) \equiv \frac{\hat{w}^2+c}{\hat{w}^2-c} \bmod n$, $\quad b = b(w) \equiv \frac{2\hat{w}}{\hat{w}^2-c} \bmod n$, where $\hat{w} \equiv w \bmod n$, if $b_1 = 1$, and $\hat{w} \equiv \frac{ws+c}{w+s} \bmod n$, if $b_1 = -1$.

Williams' Decryption. Upon receiving C the receiver firstly calculates the values $a_0 \equiv \frac{1+E(w)^2c}{1-E(w)^2c} \bmod n$, and $b_0 \equiv \frac{2E(w)}{1-E(w)^2c} \bmod n$.

The second step consists of determining $\sigma = (-1)^{b_2 - X_d(a_0)}$ and $a(w)$ and $b(w)$ by means of $a(w) \equiv \sigma X_d(a_0) \bmod n$ and $b(w) \equiv \sigma Y_d(a_0, b_0) \bmod n$.

Finally, the message w can be retrieved from $a(w)$ and $b(w)$ via

$$w = \begin{cases} \frac{a(w)+1}{b(w)} \bmod n, & \text{if } b_1 = 1, \\ \frac{cb(w)-s(a(w)+1)}{a(w)+1-sb(w)} \bmod n, & \text{if } b_1 = -1, \end{cases} \tag{2}$$

provided [4] $\gcd(b(w), n) = 1$ for $b_1 = 1$, and $\gcd(a(w) + 1 - sb(w), n) = 1$ for $b_1 = -1$.

Remark 1. By utilizing efficient methods for the combined evaluation of the Lucas sequences [24,25], it can be shown that the Williams scheme requires about twice as many multiplications as RSA, with additionally two multiplicative inverses modulo n for both encryption and decryption.

2.2 The Williams Scheme Under a CCA

Definition 1. *Let $a'(w)$ and $b'(w)$ be chosen such that $a'(w)$, $b'(w)$ correspond to $a(w), b(w)$ for the (wrong) case $b'_1 = -b_1$.*

Further denote the 'false encryption of w' by $E'(w) \equiv \frac{Y_e(a'(w),b'(w))}{X_e(a'(w))} \bmod n$, that is defined by following the formulas of the above encryption routine with respect to $b'_1 = -b_1$ (rather than b_1).

As with the Rabin scheme, the equivalence of decryption and factorization gives rise to a CCA [23]. One can even show the following [14].

Proposition 1.
- *If $\left(\frac{w^2-c}{n}\right) = 1$ or -1 then $E'(w) \equiv E(z) \bmod n$ and $b'_1(w) = b_1(z)$. Then $z \equiv D(E(z)) \bmod n$ where the parameters for the decryption routine are $b'_1(w)$ and b_2. Then $\gcd(w - z, n)$ gives the factorization of n.*
- *For $\left(\frac{w^2-c}{n}\right) = -1$ and $E'(w) \equiv E(z) \bmod n$, the problem of finding $a(z)$ for a known $a'(w)$ respectively w (and, similarly for $b(z)$) is computationally equivalent to the problem of factorizing n.*
- *If there exists an algorithm for retrieving $\pm a(w) \bmod n$ from $E(w)$ (where both values correspond to the same b_1) then there exists an efficient algorithm for factorizing n.*

2.3 Some Interesting Properties

Proposition 2. *Let b_1 be fixed and $E(w)$ as well as $a(w)$ be given. Then there is an efficient algorithm for evaluating the underlying message w.*

[4] It was shown in [14] that the number of messages not fulfilling these gcd-conditions is negligible.

Proof. For establishing this result we adopt the ideas of the attack developed in [4,9,12]. Let $A = a(w) - 1 = \frac{2c}{\hat{w}^2 - c}$ and $B = 2A^{-1} + 1$. Then $\hat{w}^2 = cB$. We may assume that $A^{-1} \bmod n$ exists.

We now consider the extension $R = \mathbb{Z}[x]/(x^2 - cB, n)$, i.e. the elements of R are polynomials of degree 1 at most with coefficients modulo n. All arithmetic operations (addition, multiplication, division) over R can be done without the knowledge of the factorization of n (where we assume that practically division is always possible). We now define a mapping $\phi : R \mapsto \mathbb{Z}_n$ by $\phi(kx+l) \equiv k\hat{w}+l \bmod n$ for $k, l \in \mathbb{Z}_n$, where according to the value b_1, $\hat{w}$ equals w, respectively $\frac{ws+c}{w+s}$. Since $\hat{w}^2 \equiv cB \bmod n$ it can easily be verified that ϕ is a ring homomorphism.

We show the result for $b_1 = -1$, since $b_1 = 1$ can be proved analogously. In particular, we then have $\phi(x - s) = \hat{w} - s = \frac{c-s^2}{w+s}$, $\phi(-xs + c) = w\frac{c-s^2}{w+s}$ and consequently $\phi(\frac{-xs+c}{x-s}) = w$. In R the expression $\frac{-xs+c}{x-s}$ becomes $\frac{c-s^2}{cB-s^2}x + \frac{cs(1-B)}{cB-s^2} \bmod n$ which we will denote by $w_1x + w_2$. Observe that, although we do not know the message w, we do know the polynomial that maps unto w, that is, w is now implicitly given by $w_1x + w_2$.

The idea behind the attack in [4,9,12] is now to encrypt this polynomial in R which gives us a polynomial in x. The homomorphic image of this encrypted polynomial then equals $E(w)$ since ϕ is a homomorphism. The combined knowledge of $E(w)$ and this homomorphic image then can be used to derive w.

To encrypt the polynomial $w_1x + w_2$ we follow the routines w.r.t. a fixed b_1. We firstly have to find the corresponding values to $a(w)$ and $b(w)$ in R. Since $\phi(x) = \hat{w}$ and $x^2 = cB$ in R, this can easily be shown to be accomplished. One obtains $a(w) \equiv \frac{B+1}{B-1}$, $b(w) \equiv \frac{2x}{c(B-1)} \bmod n$ in R.

Consequently, one evaluates the Lucas sequences w.r.t. the $a(w)$ in R modulo n and obtains the encryption in R. Let this result be denoted as $ux+v$. We stress that, since $a(w)$ and $b(w)$ in R merely consist of the public information, c, B, we know u and v. But then we know $\phi(ux + v)$ which equals $E(w)$ and therefore we have $u\hat{w} + v = E(w)$. Hence, we can solve for $\hat{w}$, if $(u, n) = 1$, which is very likely. Finally we now obtain w from $\hat{w}$ as desired. □

Remark 2. It is essential that the homomorphic image of the encrypted polynomial, which is determined by $a = a(w)$, equals $E(w)$. If a were some $a(x)$, the results obtained would be different from w. In other words, to each $E(w) = E'(z)$ correspond exactly two possible a, namely $a(w)$ and $a(z)$. Observe that from a given pair $E(w), a$ only the output z can be obtained when the factorization of n is known (cf. Proposition 1).

Proposition 3. *If* $b_1 = -1$ *and* $a(x) \equiv a(y) \bmod n$, *then* $\hat{x}^2 \equiv \hat{y}^2 \bmod n$, *where* $\hat{x} \equiv \frac{xs+c}{x+s} \bmod n$ *and* $\hat{y} \equiv \frac{ys+c}{y+s} \bmod n$.

Proof. ¿From $a(x) \equiv \frac{\hat{x}^2+c}{\hat{x}^2-c} \bmod n$ we see that $\hat{x}^2 \equiv -c\frac{1+a(x)}{1-a(x)} \bmod n$, and, analogously, for $a(y)$, $\hat{y}^2 \equiv -c\frac{1+a(y)}{1-a(y)} \bmod n$. By hypothesis the right hand sides are equal, which gives the result. □

Proposition 4. *For all* $x \in \mathbb{Z}_n^*$ *we have* $-a(x) \equiv a(c/x) \bmod n$.

Proof. Observe that $-a(x) \bmod n$ corresponds to the situation where during decryption the wrong $\sigma = \sigma(a(x))$ is obtained. By footnote 7 the decryption routine evaluates $x(-\sigma) \equiv \frac{c}{x} \bmod n$. Since also $\widehat{x}(-\sigma) \equiv \frac{c}{\widehat{x}} \bmod n$ the definition of a gives $a(x(-\sigma)) \equiv \frac{\left(\frac{c}{\hat{x}}\right)^2 + c}{\left(\frac{c}{\hat{x}}\right)^2 - c} \equiv \frac{\hat{x}^2 + c}{-\hat{x}^2 + c} \bmod n$. □

Corollary 1. *If for the case* $b_1 = -1$ *one has* $a(x) \equiv -a(y) \bmod n$ *then* $\hat{x}^2 \equiv \widehat{(c/y)}^2 \bmod n$.

3 The Proposed Scheme

3.1 Requirements on the Hash Function

Usually, semantic security is achieved via random oracles. Due to the ongoing controversy about the existence of such 'truly random' hash functions, we design our scheme in a way where we do not require the ROM. Instead, all our hash functions involved are special instances of Canetti's **oracle hash functions**. For the exact definitions we refer to [7] and only recall the fundamental concepts required for our scheme. The primitive, oracle hashing, informally describes a hash function h that, like random oracles, '*hides all partial information on its input*'.

A salient property of oracle hashing is that it cannot be deterministic, which traditionally is the case with any hash function, where two invocations on the same input yield the same answer. However, any deterministic function F is inadequate for oracle hashing, since it is bound to disclose some information on the input, as $F(x)$ itself is some information on x.

Thus, oracle hash functions need to be probabilistic in the sense that different invocations on the same input result in different outputs. The output of x is additionally determined by some *randomizer* r which is responsible for the different hash values of x. That is, the hash of x is the output of $h(x, r)$ for the random value r. Still, there needs to be some means as to verify whether a given hash value was generated from a given input x. There needs to exist a verification [5] algorithm, V, that correctly decides, given x and y, whether y is a hash of x. We use Canetti's suggestion of a **public randomness scheme**. The randomizer r appears directly in the output of $h(x, r)$. We write $h(x, r) = r, \tilde{h}(x, r)$.

The fundamental property of our underlying hash functions is Canetti's **oracle indistinguishability**. Informally, the hashes of x and y with respect to the same randomizer r, $h(x, r)$ and $h(y, r)$, should be computationally indistinguishable to any polytime adversary.

[5] The verification property is somewhat reminiscent of signature schemes. Indeed, this is exactly what will be required in our decryption verification step below. It is stressed, however, that here no secret keys are involved and all the functions can be invoked by everyone [7].

Canetti also considers the case where some (partial) information on x is already known. E.g., if for some public function f, $f(x)$ leaks some partial information on x, [6] then $\big(f(x), h(x,r)\big)$ still should be computationally indistinguishable from $\big(f(x), h(y,r)\big)$ (for details see [7], p. 467).

3.2 The Proposed Encryption and Decryption Schemes

Let $|x|$ denote the length of the string x. The concatenation of two strings x and y is denoted by $x||y$ and the bit-wise exclusive-or of x and y is denoted by $x \oplus y$. We generally use the notation $a \equiv b \bmod n$ to denote the principal remainder a, that is the unique integer $a \in \{0, ..., n-1\}$ that is congruent to b modulo n. We will assume that all calculations are carried out modulo $n = pq$. If $\overline{w}$ is the message to be encrypted let $w = 0...0\overline{w}$ be the padded message of $\overline{w}$ such that $|w| = |n|$.

Throughout, g will denote a cryptographic hash function to $\{0,1\}^{|n|}$ that is both collision resistant and pre-image resistant, while h will denote a Canetti-oracle hash function (cf. section 3.1).

The Proposed Encryption Routine $\mathcal{E} = \mathcal{E}(w)$.

1. Choose randomly a session key S and a randomizer R from $\{0,1\}^{|n|}$ such that for
$$w_R = w \oplus h(S,R), \text{ and } S_R = S \oplus h(w_R, R),$$
one has $\left(\frac{w_R^2 - c}{n}\right) = \left(\frac{S_R^2 - c}{n}\right) = -1$.
2. Calculate $a(w_R), b(w_R), E(w_R)$ and $a(S_R), b(S_R), E(S_R)$ w.r.t. $b_1 = -1$ following the routines of section 2.1.
3. Put $H = g\big(\underbrace{0...0\, a(w_R)}_{\text{length.}=|n|} \,||\, \underbrace{0...0\, a(S_R)}_{\text{length.}=|n|} \,||\, S\big)$.
4. Send the cryptogram $\mathcal{C} = [c_1, c_2, c_3, c_4] = [E(w_R), E(S_R), R, H]$.

The Proposed Decryption Routine $\mathcal{D} = \mathcal{D}(c_1, ..., c_4)$.

1. Decrypt c_1 to obtain $\sigma(w_R)a(w_R) \bmod n$, $\sigma(w_R)b(w_R) \bmod n$ following the formulas of section 2.1 for $b_1 = -1$.
2. Decrypt c_2 to obtain $\sigma(S_R)a(S_R) \bmod n$, $\sigma(S_R)b(S_R) \bmod n$ following the formulas of section 2.1 for $b_1 = -1$.
3. Select the signs σ, $\sigma(w_R) \in \{-1, 1\}$, $\sigma(S_R) \in \{-1, 1\}$ and calculate the corresponding w_R and S_R.
4. Calculate $S = h(w_R, c_3) \oplus S_R$.
5. Check whether
$$c_4 = g\big(\underbrace{0...0\, \sigma(w_R)a(w_R)}_{\text{length.}=|n|} \,||\, \underbrace{0...0\, \sigma(S_R)a(S_R)}_{\text{length.}=|n|} \,||\, S\big). \tag{3}$$

[6] It only makes sense to consider the case where f does not give full information on x. Thus, f should be one-way, or uninvertible (without the use of the secret key).

6. - If step (5) returns 'true', output $w = h(S, c_3) \oplus w_R$.
   - Otherwise goto step (3), select another sign and repeat.
   - If step (5) returns 'false' for all $\sigma(w_R) \in \{-1, 1\}$, $\sigma(S_R) \in \{-1, 1\}$ then return "NULL".

Note that b_1 is fixed. The correct values b_2 follow directly from construction, since $a(x)$ is directly in the scope of h. It can easily be seen that the signs σ of $\pm a(w_R), \pm a(S_R) \bmod n$, respectively, that pass the test in the decryption routine, are exactly the signs of the input to the hash function in the encryption routine, respectively [7]. Hence, we have

Lemma 1. *For the above routines, the decryption of an encryption of any message always gives this message.*

Remark 3.
- The testing check during decryption captures Canetti's verification property. H takes the role of the signing algorithm (with respect to the underlying w and S), and the testing step (3) takes the role of the signature verification algorithm.
- Due to the strong security properties which are achieved, some message expansion is to be expected. The entire cryptogram can be viewed as an encryption with combined signature. The hash value provides a proof of knowledge of the plaintext w and the secret parameter S. In such a setting message expansion is typical, e.g., [10,20]. More length efficient proposals have been made in [26] but the claims were not proved. This was recently done in [1] in the ROM.

4 Proof of Security

4.1 Semantic Security against Chosen Plaintext Attacks

An adversary $A = (A_1, A_2)$ defining security against CPA is usually described via the well-known game play [2]. At first, A_1 is run on input the public key, pk. At the end of $A_1's$ execution he outputs a triple (w_0, w_1, s), where w_0, w_1 are messages of the same length and s is some state information. A random one of w_0 and w_1 is selected, say w_b, and a 'challenge' y is determined by encrypting w_b under pk. A_2 is given y but not w_b. It is now $A_2's$ job to determine b, that is, to decide, if y is the encryption of w_0 or of w_1. In public key cryptography such an attack is always possible, since any adversary has access to the encryption oracle, as pk is always publicly known.

[7] Clearly, the party evaluating the hash value H can replace (the correct) $a(x)$ by $-a(x) \bmod n$ and use this as a forged hash input. Then in the deciphering process the wrong σ will be determined. In that case, it can easily be seen [14] that the (Williams)-decryption of x obtained equals $x(-\sigma) = c^{\frac{1}{x}} \bmod n$. Contrary to the forgery w.r.t. b_1 this however, does not expose the factorization of n.

Canetti showed how oracle hash functions can be used to build a crypto scheme that is semantically secure against chosen plaintext attacks [7], p. 466f. Typically, some information $f = f(x)$ is part of the cryptogram and hence establishes some public information on the secret parameter x. Canetti assumes that f is uninvertible so that this information leakage does not allow complete retrieval of x.

In our case this leads to the following technical requirement. We will assume that given $E(S_R),\ g = g\big(0...0\, a(w_R) \,||\, 0...0\, a(S_R) \,||\, S\big)$, it is impossible to find the complete underlying secret parameter S.

Remark 4. This assumption actually is not very strong. Informally, we have the following. Due to the Canetti hash function h involved, by construction no information on w_R leaks from $E(S_R)$ even if E does leak some information on S_R, where E denotes the Williams encryption of section 2.1. Also, if $a(w_R)$ did leak from $E(S_R)$ and g, then, since retrieving w_R from $a(w_R)$ is equivalent to factoring n, w_R cannot completely be recovered, so that an adversary has no information on $h(w_R, R)$. A lack of complete knowledge of $h(w_R, R)$ implies a lack of complete knowledge of S, even if S_R could completely be recovered from $E(S_R)$ and g. Similarly, if some partial information on S_R and w_R can be obtained by the combined knowledge of $E(S_R)$ and g, again by the Canetti-hash function h, S cannot completely be recovered. Thus, S would need to leak in full from g to violate our assumption.

Analogously as in [7], we obtain the semantic security of the proposed scheme.

Theorem 1. *The proposed cryptosystem is semantically secure against adaptive chosen plaintext attacks, if the factorization of $n = pq$ is hard, h is a Canetti oracle hash function with the additional technical assumption above on the cryptographic function g.*

Proof. (Sketch) Assume an adversary $\mathcal{A}$ that does break the scheme under a CPA. Let the probability for his success be as defined in the proof to Theorem 10 in [7]. The tuple $E(S_R),\ g = g\big(0...0\, a(w_R) \,||\, 0...0\, a(S_R) \,||\, S\big)$, yields some information f on S which by the assumption above corresponds to the uninvertible function f in Canetti's case.

Construct an algorithm $\mathcal{D}$ that distinguishes between $(f(S), h(S, R))$ and $(f(S), h(S', R))$, where S, S', R are randomly chosen and $f(S) = (E(S_R), g)$. Since R is public, $h(S, R) = R, \tilde{h}(S, R)$, and by the requirement that h is a Canetti hash function, it follows that for uniformly chosen S, R the value $h(S, R)$ is uniform in $\{0, 1\}^l$ for some l.

Given $f(S), R, \xi$ (where ξ is either $\tilde{h}(S, R)$ or $\tilde{h}(S', R)$), the distinguisher $\mathcal{D}$ will construct a ciphertext in the following way. $\mathcal{D}$ may choose either one of w_0 or w_1 as message in the game play defining security against CPA. Assume that he chooses w_1. Then he obtains $w_R = w_1 \oplus \xi$ and he can hand $\mathcal{A}$ the 'ciphertext' $C = [E(w_R), E(S_R), R, g]$. Now, if $\mathcal{A}$ outputs 'w_1' then $\mathcal{D}$ outputs '$\xi = \tilde{h}(S, R)$'. Otherwise $\mathcal{D}$ outputs '$\xi = \tilde{h}(S', R)$'.

As in Canetti's case this follows since in the former event the constructed w_R was the correct one, while in the latter, it must must have been equal to

$w_R = w_1 \oplus \tilde{h}(S', R) \neq w_1 \oplus \tilde{h}(S, R)$. In particular, then $\mathcal{A}$ is given an encryption of a uniformly chosen message. The decryption cannot be w_1, hence in that case, by the CPA game, it can only be w_0, which $\mathcal{A}$ outputs.

Analyzing $\mathcal{D}$ is straightforward with the exact success probability given in [7]. The existence of such a distinguisher yields a contradiction to the assumption. □

4.2 'Weak-Sole-Samplability'

Recall the notion of a valid ciphertext. This is such where the decryption oracle does not reject. We now completely characterize all possibilities how for the proposed scheme valid ciphertexts can be obtained.

The randomizer R, since it directly occurs in C, plays a unique role. Nonetheless, this information cannot be used for any attack. (Compare also Canetti's discussion on this public randomizer [7]).

Lemma 2. *Let $\mathcal{C} = [c_1, c_2, c_3, c_4]$ be a valid ciphertext, h a Canetti oracle hash function, g a cryptographic hash function, and suppose that the factorization of n is infeasible. If in $\mathcal{C}$ the c_3 gets modified, then a necessary condition for obtaining another valid ciphertext is that all entries in $\mathcal{C}$ get modified.*

Proof. We analyze any adversary that tries to obtain another valid ciphertext. Let c_3' be the modified value and let $\mathcal{C}$ be the encryption of the message w relative to the session key S and the randomizer R. We can assume that the adversary knows w (e.g. by mounting a CCA). We can also assume that he knows S (e.g. by his own encryption) because otherwise any such attack would not be possible (this follows from the fact that c_4 remains unchanged, g is both collision resistant and pre-image resistant and since the given C is a valid ciphertext). By their definition he then also knows w_R and S_R.

Suppose firstly that $\mathcal{C}' = [c_1, c_2, c_3', c_4]$ is also valid. Then the validity check (3) passes if $c_4 = g(...||S) = g(....||S')$, where $S' = S'(c_3')$ is in the fourth step of the deciphering oracle computed as $S' = h(\sigma(w_R)w_R, c_3') \oplus S_R$. By the choice of h necessarily $S' = S$ so that $h(\sigma(w_R)w_R, c_3')$ has to evaluate to $S \oplus S_R$. But that would imply that $h(w_R, c_3) = h(w_R, c_3')$ which is extremely unlikely [7].

Similarly, we see that $\mathcal{C}' = [c_1, c_2', c_3', c_4]$ where c_2' is determined a priori, leads to a contradiction. Consequently, the adversary needs to evaluate a modified c_2 accordingly, i.e. such that the properties of the hash function are not being violated. This is only possible if at first the hash input, that is some c_3', is being selected. As above, we again need to have $S' = S$, where now $S' = h(w_R, c_3') \oplus D(c_2')$, and D is the Williams decryption of section 2.1. ¿From the hash output and $S' = S$ the adversary then obtains the decrypted value $x = D(c_2')$ (w.r.t. $b_1 = -1$) of the forged c_2', that is, $x = S_R'$ (respectively c/S_R').

However, this x has to be of a special form (this will lead to the contradiction below), because in the validity check it is required that

$$c_4 = g(...||0..0\,\sigma(S_R)a(S_R)||...) = g(...||0..0\,\sigma(x)a(x)||...)$$

(where we assume that the hash input is split up according to the appropriate lengths).

By Proposition 4 the above identity is only possible if either $a(S_R) \equiv a(x)$ mod$n$, or $a(S_R) \equiv a(c/x) \bmod n$, depending on whether the above σ's correspond or not. According to Proposition 3 and Corollary 1,

$$\text{either } \widehat{S_R}^2 \equiv \hat{x}^2 \bmod n \text{ or } \widehat{S_R}^2 \equiv \widehat{(c/x)}^2 \bmod n.$$

But we also have that $c_2 \not\equiv c_2' \bmod n$. Also, by assumption, both $\mathcal{C}$ and $\mathcal{C}'$ are valid which means that the test passes for exactly one $\sigma(S_R)$ and thus for exactly one $\sigma(x)$ which then yields the corresponding values, x or c/x, respectively.

Since the decryption of c_2', as well as that of c_2, is being conducted w.r.t. $b_1 = -1$, the preimages, x and S_R, respectively c/x and S_R, need to be distinct mod n. Then also $\hat{x}$ and $\widehat{S_R}$, respectively $\widehat{c/x}$ and $\widehat{S_R}$ need to be distinct since otherwise $s^2 \equiv c \bmod n$, contrary to the choice of s.

Further, we can show that $x \not\equiv -S_R$, respectively $c/x \not\equiv -S_R \bmod n$. These two cases can be dealt with the same way. Observe that x was defined according to the hash output of c_3', i.e. as $x = h(w_R, c_3') \oplus S$. If we assume that $x \equiv -S_R \bmod n$ then c_3' (which has been selected a priori) would hash to the specific output $S \oplus (-S_R) \bmod n$, a contradiction. Analogously, $\hat{x} \not\equiv -\widehat{S_R}$, respectively $\widehat{c/x} \not\equiv -\widehat{S_R} \bmod n$. But then $\gcd(\hat{x} - \widehat{S_R}, n)$, respectively $\gcd(\widehat{c/x} - \widehat{S_R}, n)$ is a proper factor of n. To find this factor the adversary only needs to know $\hat{x}$, respectively $\widehat{c/x}$ and $\widehat{S_R}$, which he does when he knows x.

Observe that the adversary already knows w, w_R and S_R. However, Proposition 2 asserts that the adversary can calculate x from $E(x) = c_2'$ and $a(x) \equiv a(S_R)$ respectively $-a(S_R) \bmod n$.

Thus, the adversary would find the factorization of n. The derived contradiction to the hypothesis of the lemma implies that the adversary cannot compute a valid ciphertext by just forging c_2 and c_3.

The adversary can also try to forge c_1. But, in order to pass the test, then he would need to know c_1' along with the corresponding $\sigma(y)a(y)$, where y (respectively c/y) is the decryption of c_1' (w.r.t. $b_1 = -1$).

As decrypting c_1' or determining this $a(y)$ is equivalent to factoring (Proposition 1), the adversary can only, conversely, define y as w_R' and encrypt y (w.r.t. $b_1 = -1$) to obtain his forged c_1'. Similarly as above, he needs to evaluate S_R' as $h(w_R', c_3') \oplus S$ in order to fulfill the requirement on the hash function in (3) with respect to the last block in the input. But this now constitutes a special form of the attack considered above. The adversary would have to forge c_2 which is impossible, independent of the choice of c_1. □

Let us consider an adversary that has access to g, h, $\mathcal{E}$, and $\mathcal{D}$. He can play with his encryption oracle, and may also make t queries of the decryption oracle. He then produces a new valid ciphertext that he outputs. As in [2] we demand that the adversary never outputs a string that coincides with the value returned from some $\mathcal{E}$-query.

The basic idea in both Lemma 2 and Theorem 2 below is to analyze the different possibilities as how an attacker might be able to reuse existing valid ciphertexts. That is, we investigate all ways for obtaining valid ciphertexts (other than running the encryption oracle).

We will give a complete characterization of all possibilities to find a valid ciphertext. Depending on whether the adversary knows the secret parameter S corresponding to some known valid ciphertext, he may follow only one of the specific steps given in the proof below. In each of these particular cases the proof also shows that the adversary is not able to generate any new valid ciphertext whose plaintext he does not know.

For $1 \leq i \leq t$, let $\mathcal{C}_i = [c_1^{(i)}, c_2^{(i)}, c_3^{(i)}, c_4^{(i)}]$ be the ith valid cryptogram that the adversary gets decrypted. Let $\mathcal{C}'$ be the new valid ciphertext that the adversary produces. By Lemma 2 we only need to distinguish between the following types of attacks.

- Type I: There is some $1 \leq j \leq t$ such that for $\mathcal{C}_j = [c_1, c_2, c_3, c_4]$,
  (a) $\mathcal{C}' = [c_1' \neq c_1, c_2, c_3, c_4]$, (b) $\mathcal{C}' = [c_1, c_2' \neq c_2, c_3, c_4]$,
  (c) $\mathcal{C}' = [c_1' \neq c_1, c_2' \neq c_2, c_3, c_4]$,
- Type II: There is some $1 \leq j \leq t$ such that for $\mathcal{C}_j = [c_1, c_2, c_3, c_4]$, (a) $\mathcal{C}' = [c_1, c_2, c_3, c_4' \neq c_4]$, (b) $\mathcal{C}' = [c_1' \neq c_1, c_2, c_3, c_4' \neq c_4]$,
  (c) $\mathcal{C}' = [c_1, c_2' \neq c_2, c_3, c_4' \neq c_4]$, (d) $\mathcal{C}' = [c_1' \neq c_1, c_2' \neq c_2, c_3, c_4']$,
- Type III: For all i, $\mathcal{C}' = [c_1' \neq c_1^{(i)}, c_2' \neq c_2^{(i)}, c_3' \neq c_3^{(i)}, c_4' \neq c_4^{(i)}]$.

Theorem 2. *Assume that h is a Canetti oracle hash function, g is a cryptographic hash function, and that it is computationally infeasible to find the factorization of n. Then the above encryption scheme is weak-sole-samplable. Any valid cryptogram that is not an $\mathcal{E}$ output, has to be the result of some type II or III attack with the individual steps described below. In both cases, the adversary then knows the underlying w, S, as well as the underlying signs σ in the hash-input.*

Proof. **Type I attacker:**
Suppose we have a type I (a) attacker. Because c_4 is fixed we can as in the proof to Lemma 2 assume that the attacker knows the corresponding w and S. Observe that, since $\mathcal{C}$ is valid, the $S = h(w_R, c_3) \oplus S_R$ obtained in the fourth step of $\mathcal{D}$ passes the test (3) for the unique $\sigma(w_R)$ and thus for the unique w_R. If now $c_1 \neq c_1'$ then the S' obtained will be different from S. This follows, since for fixed $b_1 = -1$ the decryption of c_1' is either w_R' or c/w_R'. These values are different from w_R because otherwise $c_1 = c_1'$. Therefore the test will reject for this S'. In order to obtain the same S, also c_2 would have to be modified, which is not the case under the type of attack under inspection.

Similarly we see that a type I (b) attack will be rejected by the test because the S' obtained in step (4) will not match the valid S.

Now consider a type I (c) attacker. In order to guarantee that the S' obtained in step (4) equals the valid S, the adversary can only proceed analogously as

in the proof to Lemma 2. He needs to define the (Williams) decryption of the modified c_2', that is S_R', as $S \oplus h(y, c_3)$, where $y = w_R'$ is the decryption of c_1'. But these w_R' and S_R' need to pass (3). Similarly as in Lemma 2, he would be able to factorize n, a contradiction. Hence, any type I attack will get rejected as well.

Type II attacker:
For a type II (a) attacker observe that by definition c_1, c_2, c_3 remain unchanged. Hence, in steps 2 and 3 during decryption, the quantities $\pm a(w_R)$, $\pm a(S_R)$ corresponding to the original w and S are obtained. Since $c_4 = g(\sigma(w_R)a(w_R) ||\sigma(S_R)a(S_R)||S)$ for the specific $\sigma(w_R)$, $\sigma(S_R)$, one can only obtain a modified hash output w.r.t. different signs, $\sigma(w_R)$ and/or $\sigma(S_R)$. The requirement on g necessitates that the adversary knows the individual blocks in the hash input (he can only obtain the output from the input). As he also needs to know c_1 and c_2, by Lemma 2, he knows the modified message w as well as the modified S that result in the modified cryptogram due to the change of the σ's.

In a type II (b) attack the test only passes if the hash output, c_4' is calculated as the hash-output w.r.t. the modified c_1'. Then the adversary has to know the $\sigma(w_R')a(w_R')$ that is obtained by decrypting c_1'. As usual, by Lemma 2, we conclude that he can find w_R', respectively c/w_R'. To obtain the hash value c_4' he also needs to know the $\sigma(S_R)a(S_R)$. Again, since he knows c_2 he then knows S_R. Depending on the σ's selected he obtains two different (modified) S and four different (modified) w. He can easily verify which of those have the desired encryptions c_1' so that he knows the modified w and S that result in $\mathcal{C}'$.

Exactly the same way we can show that in a type II (c) attack the adversary needs to know the underlying quantities that result in $\mathcal{C}'$.

A type II (d) attack can be dealt with analogously, because knowledge of $a(x)$ and $E(x)$ is equivalent to knowing x, where x firstly is w_R' and then secondly S_R'.

Type III attacker:
The result follows exactly as for a type II (d) attacker because the value c_3 is not essential. The adversary would need to know the first two blocks of the input to the hash function. Along with c_1' and c_2' this is equivalent to knowing w_R' and S_R' where $c_3' = R'$. However, since c_3' is public one easily finds the underlying w' and S' from the randomized w_R' and S_R'.

We have shown that valid ciphertexts cannot be obtained apart from knowing their underlying parameters, which completes the proof of Theorem 2 in all cases. □

Acknowledgements

I am deeply grateful to the following people for their valuable comments and for their support: Dr. A. Desai, Professors A. Menezes, W.B. Müller, D. Pointcheval, P. Rogaway, C.P. Schnorr, and N. Smart. Also, I would like to thank the referees for their careful reading of the manuscript and for their insightful and helpful remarks.

References

1. J. Baek, B. Lee, K. Kim, Provably Secure Length-saving Public-Key Encryption Scheme under the Computational Diffie-Hellman Assumption. *ETRI J.*, Dec. 2000.
2. M. Bellare, A. Desai, D. Pointcheval and P. Rogaway, Relations among notions of security for public-key encryption schemes, Extended abstract in *Advances in Cryptology - Crypto 98, LNCS*, 1462, H. Krawczyk (ed.), Springer (1998); full version available at `www-cse.ucsd.edu/users/mihir/papers/crypto-papers.html`.
3. M. Bellare, P. Rogaway, Optimal asymmetric encryption – How to encrypt with RSA, *Advances in Cryptology - Eurocrypt 94, LNCS* 950, A. De Santis (ed.), Springer (1995) pp. 92–111.
4. D. Bleichenbacher, On the Security of the KMOV Public Key Cryptosystem, *Advances in Cryptology - Crypto'97, LNCS* 1294, Springer (1997) pp. 235 – 248.
5. D. Bleichenbacher: Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1. *Adv. in Cryptology - Crypto'98, LNCS* 1462, H. Krawczyk (ed.), Springer (1998) pp. 1–12.
6. D. Boneh, R. Venkatesan, Breaking RSA May Not Be Equivalent to Factoring, *Advances of Cryptology - Eurocrypt '98, LNCS* 1403, K. Nyberg (ed.), Springer (1998) pp. 59–71.
7. R. Canetti, Towards Realizing Random Oracles: Hash Functions That Hide All Partial Information, *Advances in Cryptology - Crypto'97*, 455-469.
8. R. Canetti, O. Goldreich, S. Halevi, The random oracle model, revisited, In: *30 th Annual ACM Symp. on Theory of Computing* (1998).
9. D. Coppersmith, M. Franklin, J. Patarin, M. Reiter, Low-Exponent RSA with Related Messages, *Advances of Cryptology - Eurocrypt' 96, LNCS* 1070, U. Maurer (ed.), Springer (1996) pp. 1–9.
10. R. Cramer, V. Shoup, A Practical Public Key Cryptosystem Provable Secure against Adaptive Chosen Ciphertext Attack, *Advances of Cryptology - Crypto '98, LNCS* 1462, H. Krawczyk (ed.), Springer (1998) pp. 13–25.
11. D. Dolev, C. Dwork, M. Naor, Non-malleable cryptography, In *23rd Annual ACM Symp. on Theory of Computing,* (1991) pp. 542–552.
12. R. Gennaro, A. Shamir, Partial Cryptanalysis of Koyama's Eurocrypt'95 scheme, LCS Technical Memo 512, May 10 (1996) MIT.
13. S. Goldwasser, S. Micali, Probabilistic Encryption, *Journal of Computer and System Sciences* 28 (April 1984) pp. 270–299.
14. S. Müller, Some Observations on Williams General Encryption Scheme, Some Remarks on Williams' Public Key Crypto Functions, Manuscripts, submitted, University of Klagenfurt (2000).
15. M. Naor, M. Yung, Public-key cryptosystems provably secure against chosen ciphertext attacks, In *22nd Annual ACM Symp. on Theory of Computing,* (1990) pp. 427–437.
16. T. Okamoto, S. Uchiyama, E. Fujisaki: EPOC: Efficient Probabilistic Public-Key Encryption, submission to P1363a (1998).
17. D. Pointcheval, Chosen-Ciphertext Security for any One-Way Cryptosystem, *PKC'2000*, H. Imai, Y. Zheng (eds.), Springer (2000).
18. M. O. Rabin: Digitalized signatures and public-key functions as intractable as factorization. MIT/LCS/TR-212, MIT Laboratory for Computer Science, 1979.
19. C. Rackoff, D. Simon, Non-interactive zero-knowledge proofs of knowledge and chosen ciphertext attack, *Advances in Cryptology - Crypto'91, LNCS*, 576, Springer (1991).

20. C.P. Schnorr, M. Jakobsson, Security of Signed ElGamal Encryption, To appear, Asiacrypt'00.
21. V. Shoup, Using Hash Functions as a Hedge against Chosen Ciphertext Attack, `http://philby.ucsd.edu/cryptolib/1999.html` (1999).
22. Y. Tsiounis, M. Yung, On the security of ElGamal-based encryption, *PKC'98*, LNCS 1431, Springer (1998), pp. 117-134. `www.ccs.neu.edu/home/yiannis/pubs.html`.
23. H.C. Williams, Some Public-Key Crypto-Functions as Intractable as Factorization, *Cryptologia* 9 (1985) pp. 223-237.
24. H.C. Williams, A $p+1$ method of factoring. *Math. Comp.* **39**, no. 159 (1982) pp. 225–234.
25. H.C. Williams, "Édouard Lucas and Primality Testing", Canadian Mathematical Society Series of Monographs and Advanced Texts, Vol. 22 (1998), John Wiley & Sons.
26. Y. Zheng, J. Seberry, Immunizing public key cryptosystems against chosen ciphertext attacks, *IEEE Journal on Selected Areas in Communications*, Vol. 11, No. 5 (1993) pp. 715–724.