On the Security of a Williams Based Public Key Encryption Scheme[1]

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

January 15, 2000

[1]Research supported by the Österreichischen Fonds zur Förderung der wissenschaftlichen Forschung, FWF-Project no. P 13088-MAT.

Abstract

Motivated by some strong and interesting cryptographic properties of the intrinsic structure of Williams' encryption scheme [36], we present a practical modification thereof that can be proven secure against adaptive chosen-ciphertext attacks. While our proof applies a notion of security closely related to the strong concept of 'plaintext-awareness' [4], we do not require the use of random oracles. To this end, we introduce a comparable notion and prove the security without relying on the ROM. In particular, our underlying intractability assumptions are the factorization problem of any large integer $n = pq$, and a concept similar to the "oracle hashing" paradigm introduced in [11]. The main advantage of our system is that we do not rely on any special form of the modulus $n = pq$, nor do we require any specific values of the primes p and q. Furthermore, we do not rely on "truly random" hash functions. The assumptions that we make are both well-defined and reasonable.

Keywords: Chosen Ciphertext Security, Message Awareness, Factorization Intractability, Oracle Hashing, Williams' Encryption Scheme

Contents

1 Introduction

1.1 Provably Secure Systems

A desirable property of any cryptoscheme is a proof that breaking it is as difficult as solving a computational problem that is widely believed to be difficult, such as integer factorization or the discrete logarithm problem. Consequently, a cryptographic scheme is *provably secure* if an attack on the scheme implies an attack on the underlying primitives it employs.

While RSA [29] is undoubtedly the most well-known and widely used public-key cryptosystem, it is not known if breaking RSA is as difficult as factoring. The question of whether knowledge of the factors of the modulus n is required in order to invert RSA, remains open (cf. [7, 8]) because we know of attacks that do not rely on the factorization intractability. It is therefore of interest to develop cryptographic schemes whose security is equivalent in the difficulty to factoring the modulus, i.e. for which it is necessary to factor n in order to retrieve plaintext from ciphertext without using the secret key. A variety of factorization equivalent RSA modifications have been proposed (cf. [18, 20, 22, 30, 31, 35, 36, 37]) which are essentially based on the same underlying idea. By constructing specific en- and decryption functions, the decrypter will not simply obtain the original message, but rather a variety of possible solutions. As a result, the encrypter needs to provide a clue indicating which of these ambiguously decoded messages is the correct one. Upon transmitting some other information than the one which would help to identify the correct message, the sender can manipulate the decoder to decrypt a 'wrong message'. This decrypted message then can be used by the sender to factorize the modulus in a few additional steps. Because of this problem, all these provable secure systems, since they are based on a similar constructive nature, will be vulnerable to a ***chosen ciphertext attack*** (CCA). Under such an attack the adversary selects the ciphertext and is then given the corresponding plaintext. However, since it is largely agreed upon that security against CCA is one of the most important attributes of any public key cryptosystem, the main goal of this paper consists of establishing a factorization equivalent encryption scheme which is simultaneously secure against such an attack.

1.2 Chosen Ciphertext Secure Systems

• **Theoretical CCA secure schemes:** Provably secure cryptosystems secure against CCA were first given in [13, 25, 27]. Unfortunately, these schemes rely on theoretical constructions of non-interactive zero-knowledge proofs, and as such are quite impractical.

• **Practical CCA secure schemes:** Practical cryptosystems intended to be secure against chosen ciphertext attack were first initiated by Damgård [12] and further extended by Zheng and Seberry [39, 40, 41]. However, these schemes do not address the issue of provable security, in that no proofs based on known intractability assumptions are presented.

Motivated by the problem of establishing provable security in terms of an underlying intractability assumption, Bellare and Rogaway [1, 2, 3], presented practical and provably secure schemes in the ***random oracle model*** (ROM). They suggest first designing cryptographic protocols in an *ideal system*, and then replacing oracle access by the computation of an "appropriately chosen" function. Under the strong assumption about the underlying cryptographic hash function being as good as an "ideal" random-oracle their methods have been validated to be semantically secure. They present two random-oracle based encryption schemes by utilizing the RSA, respectively the Diffie-Hellman, primitive. Another ElGamal based instantiation, that also relies on the random oracle model, was presented by Tsiounis and Yung [34].

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, Canetti, Goldreich, and Halevi [10] prove that there are encryption and signature schemes which are secure in the ROM, but have no secure implementation (replacement of the random oracle by any easy to evaluate function) in the "real world" (where a random oracle does not exist).

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 designing a chosen ciphertext secure system which is practical and ***proven secure under well defined intractability assumptions***. The first encryption scheme of this kind was established by Cramer and Shoup [9, 32]. Their proof of security relies only on the hardness of the Diffie-Hellman decision problem and the collision intractability of a hash function.

1.3 The Proposed Scheme

The contribution of this paper is a CCA *secure modification of Williams' encryption scheme* [36] that is based on the factorization problem of any large number, and some well defined and reasonable properties of a cryptographic hash function. Our proposal has the following advantages:

- **A CCA secure encryption scheme based on factoring** $n = pq$: It seems to be widely agreed upon that the Diffie-Hellman problem on the one hand, and the factorization problem on the other hand provide two computationally hard cryptographic primitives. As noted above, a number of cryptosystems have been presented that are based on Diffie-Hellman without having to require random oracles. But no CCA secure encryption scheme has been published yet that utilizes the factorization utility of arbitrary numbers in the standard model. Very recently, a submission has been made [26] of an encryption scheme whose security relies on the ROM, and additionally requires the very specific structure of the modulus, $n = p^2q$. The first attempt of realizing security of moduli of the form $n = pq$ has been presented in [23] where $p \equiv q \equiv 3 \bmod 4$. However, based on some misunderstanding of some results in an earlier version of [17], the security of [23] is only achievable in the ROM. By contrast, our proposed scheme of this paper overcomes these limitations.

- **No special form of the primes**: We stress that our securing method in an analogous way could be applied to the factorization equivalent RSA version of [35] which utilizes a modulus $n = pq$ of the form $p \equiv 3 \bmod 8$, $q \equiv 7 \bmod 8$. Although no algorithm is known that exploit this special structure of the primes for more efficiently factoring n, we present our CCA securing method for the most general form of the RSA modifications, presented by H.C. Williams in [36]. While en- and decryption of the Williams scheme can quite efficiently be realized by means of evaluation methods of combined Lucas sequences (cf. [38]), the main focus of this scheme has never been extremely high practicality.

  However, the major advantage of the Williams system [36] relies on the fact that it does not rely on any specific structure of the prime factors of n. In fact, it is the only factorization equivalent RSA modification known that does not require a specific form of the modulus. Consequently, since we base our system on Williams' general scheme, we also require no special form of the primes.

- **Reasonable and well defined properties of the hash function**: The proof of security of our system relies on the factorization primitive and on *well defined* assumptions on the hash function. In detail, we rely on the existence of a "*suitable*" *hash function* (similar as in [11, 14]) and on a `generalization of the plaintext-awareness paradigm` defined in [2, 3, 4].

  It is shown that a specific, message dependent information, that is included as part of the cryptogram, does identify any attempted CCA or confirms the validity of the cryptogram being sent. The decryption oracle will only return a useful (i.e. non NULL) output to cryptograms whose underlying messages are known to the sender. The party which is able to generate a 'valid ciphertext' indeed also has to know the corresponding message and consequently cannot abuse the decryption system into outputting 'wrong' messages as it must have known the result of its query to begin with. As a result, the attacker will not learn any new information by having access to the decryption oracle.

  The idea is very similar to plaintext-awareness (PA) concept of [2, 3, 4]. However, their definition is *only* achievable in the random-oracle model. Indeed, the authors of [4] give reasonable arguments why their concept would not apply in a scheme not using random oracles (p. 24 of full version of [4]). They even post it as an interesting open question to find an analogous formulation of PA in the standard model.

  The main goal of this paper thus consists in defining such suitable generalization and showing that the results relating to the achieved security are also true in a ROM devoid model.

As mentioned above, our main motivation for applying the Williams scheme as basis for the security enhanced method, is to be able to utilize the general factorization intractability of any numbers $n = pq$. Moreover, it will turn out that the Williams system actually already incorporates very strong cryptographic properties, which in a very natural way can be exploited for realizing security against CCA. Our proof of security will heavily rely on the congruence $x^2 \equiv y^2 \bmod n$ with $x \not\equiv \pm y \bmod n$. This property, however, is already intrinsically a part of the Williams scheme (cf. e.g. Proposition 6.5 below). Not only is the Williams scheme the only RSA generalization that is as strong as factoring, but also it naturally seems to provide suitable properties for simultaneously achieving security against CCA.

2 Description of the Underlying Williams' System

We will give a short description of some of the fundamentals of Williams' scheme [36], which essentially will be the basis for our proposed CCA secure system.

2.1 Key Generation and Fundamental Properties

First, two large primes p and q are selected by the designer and their product, n, is computed. Then, a value c is chosen such that $p \equiv -\left(\frac{c}{p}\right) \bmod 4$ and $q \equiv -\left(\frac{c}{q}\right) \bmod 4$. Additionally, a value s with $\gcd(s,n) = 1$, $\gcd(s^2 - c, n) = 1$ and $\left(\frac{s^2-c}{n}\right) = -1$ is determined. We shall assume throughout that the parameters p, q, n, c, and s are chosen according to the above constrains. 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 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.* Let

$$\begin{cases} b_1 = 1, \text{ and } \gamma(w) = w + \sqrt{c}, & \text{if } \left(\frac{w^2-c}{n}\right) = 1, \\ b_1 = -1, \text{ and } \gamma(w) = ws + c + (w+s)\sqrt{c}, & \text{if } \left(\frac{w^2-c}{n}\right) = -1. \end{cases} \tag{1}$$

Of special interest for the cryptosystem are numbers of the form $\frac{\gamma(w)}{\overline{\gamma}(w)} \equiv a(w) + b(w)\sqrt{c} \bmod n$, where $\overline{\gamma}(w) = w - \sqrt{c}$, respectively $(w - \sqrt{c})(s - \sqrt{c})$, according as $b_1 = 1$ or -1. For a given w are the parameters $a(w)$ and $b(w)$ uniquely determined modulo n, as can be seen from the following proposition (cf. [36]).

Proposition 2.1. *Suppose* $\gcd(w^2 - c, n) = 1$ *and* $\alpha(w) = \frac{\gamma(w)}{\overline{\gamma}(w)} \equiv a(w) + b(w)\sqrt{c} \bmod n$. *Then*

$$\begin{aligned} a(w) &\equiv \frac{\hat{w}^2 + c}{\hat{w}^2 - c} \bmod n, \\ b(w) &\equiv \frac{2\hat{w}}{\hat{w}^2 - c} \bmod n, \end{aligned} \tag{2}$$

where

$$\begin{cases} \hat{w} \equiv w \bmod n, & \text{if } b_1 = 1, \\ \hat{w} \equiv \frac{ws+c}{w+s} \bmod n, & \text{if } b_1 = -1. \end{cases}$$

We will throughout use the notation of Proposition 2.1.

The system is essentially based on the identity

$$\alpha^{2ed} \equiv \pm\alpha \bmod n. \tag{3}$$

For general α of the form $a + b\sqrt{c}$ this congruence usually will not hold. In [36] this difficulty is overcome by constituting special en- and decryption functions, each basically consisting of two parts. The special form of the en- and decryption will guarantee the validity of (3).

In summarizing Williams' results, the fundamental principle may be characterized in the following way.

Lemma 2.2. *Let $w \in \mathbb{Z}_n$ such that* $\gcd(w^2 - c, n) = 1$. *If $\alpha(w) \equiv a(w) + b(w)\sqrt{c} \bmod n$, where $a(w)$ and $b(w)$ are determined according to the corresponding case for b_1 in Proposition 2.1, then $a(w)^2 - cb(w)^2 \equiv 1 \bmod n$, $\left(\frac{2(a(w)+1)}{n}\right) = 1$ and $\alpha(w)^{2ed} \equiv \pm\alpha(w) \bmod n$.*

The en- and decryption functions are based on the following sequences.

Definition 2.3. For $\alpha = \alpha(w), a = a(w), b = b(w)$ of Proposition 2.1 define the sequence of integers

$$X_i(\alpha) = X_i(a) = \tfrac{\alpha^i + \overline{\alpha}^i}{2} = \tfrac{V_i(2a,1)}{2} \tag{4}$$
$$Y_i(\alpha) = Y_i(a,b) = b\tfrac{\alpha^i - \overline{\alpha}^i}{\alpha - \overline{\alpha}} = bU_i(2a,1), \tag{5}$$

where U_i, respectively V_i, denote the Lucas sequences of the first and second kind, respectively (cf. [28]).

Most interestingly, the powers of α and $\overline{\alpha}$ correspond to the terms of the X_i- and Y_i- functions by the characterization

$$\alpha^i = X_i(\alpha) + Y_i(\alpha)\sqrt{c}, \qquad \overline{\alpha}^i = X_i(\alpha) - Y_i(\alpha)\sqrt{c}. \tag{6}$$

2.2 The Encryption and Decryption Routines of [24]

Theorem 3.1 in [36] poses some restrictions on the message w that is to be encrypted. Although for sufficiently large numbers this is not a real problem [24], it is shown in [24] that a simple variation of the Williams scheme minimizes these restrictions. The modification to [36] essentially consists of reversing the original encryption function $X_e(a(w)) \cdot {Y_e}^{-1}(a(w), b(w)) \bmod n$ and adapting the decryption scheme. In the following we will exclusively be applying this slightly modified Williams' scheme.

Encryption E**:** The first step of the encryption process consists of calculating $a(w), b(w)$ from the message w by means of Proposition 2.1. Then w is encoded as

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

The cryptotext C to be transmitted is the triple $[E(w), b_1, b_2]$, where b_1 is defined in (1) and b_2 equals 0 or 1, depending on whether $a(w) \bmod n$ is even or odd.

Decryption D**:** Upon receiving C the receiver firstly calculates the values

$$a_0 \equiv \frac{1+E(w)^2c}{1-E(w)^2c} \bmod n, \qquad b_0 \equiv \frac{2E(w)}{1-E(w)^2c} \bmod n.$$

The second decryption step consists of determining $\sigma = (-1)^{b_2 - X_d(a_0)}$ and $a(w), b(w)$ by means of

$$a(w) \equiv \sigma X_d(a_0) \bmod n \text{ and } b(w) \equiv \sigma Y_d(a_0, b_0) \bmod n.$$

Finally, the message w can be retrieved from $a(w)$ and $b(w)$ by means of

$$\tilde{w} \equiv \frac{a(w)+1}{b(w)} \bmod n, \text{ and}$$

$$\begin{cases} w \equiv \tilde{w} \bmod n, & \text{if } b_1 = 1, \\ w \equiv \frac{\tilde{w}s-c}{s-\tilde{w}} \bmod n, & \text{if } b_1 = -1, \end{cases} \tag{7}$$

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

It was further shown in [24] that the number of cases where the above constrains on $a(w)$ and $b(w)$ in the decryption routine are not fulfilled, are negligible.

3 The Williams Scheme Under the CCA

3.1 Some Fundamentals

In [36] Williams proved that decryption of messages without the use of the secret key leads to the factorization of n. While this result establishes provable security with respect to the factorization primitive, by the same token it implies that the scheme succumbs to a CCA. For both schemes, [36] and [24], an essential step for demonstrating the factorization equivalence (respectively for making a CCA possible) consists of assigning w 'wrong' parameters for $a(w)$ and $b(w)$ which give rise to a 'wrong encryption' of w.

Definition 3.1. Let $a'(w)$ and $b'(w)$ be chosen such that

$$\begin{cases} \text{for } b_1 = 1 & a'(w), b'(w) \text{ correspond to } a(w), b(w) \text{ for } b_1 = -1, \\ \text{for } b_1 = -1 & a'(w), b'(w) \text{ correspond to } a(w), b(w) \text{ for } b_1 = 1. \end{cases}$$

Similarly, let $\gamma'(w)$ be defined as $\gamma(w)$ for $b'_1 := -b_1$. If, further $\left(\frac{w^2-c}{n}\right) = -1$ we 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' (rather than b_1).

By means of (1) and Proposition 2.1 it becomes obvious that also $\alpha'(w) \equiv \frac{\gamma'(w)}{\overline{\gamma'(w)}} \equiv a'(w) + b'(w)\sqrt{c} \bmod n$.

The crucial point for establishing the equivalence to factoring is the following proposition (cf. Lemma 4.3 of [36]).

Proposition 3.2. *If* $\left(\frac{w^2-c}{n}\right) = -1$ *then there exist exactly four values of* z *modulo* n*, such that* $(w-z)(wz-c) \equiv 0 \bmod n$*. If* $z \not\equiv w \bmod n$ *then there are two cases.*

(i) $z \equiv w \bmod p, z \equiv \frac{c}{w} \bmod q$ *or* $z \equiv w \bmod q, z \equiv \frac{c}{w} \bmod p$*. Then* $\left(\frac{z^2-c}{n}\right) = 1$ *and* $(z-w, n)$ *is a proper factor of* n*.*

(ii) $z \equiv \frac{c}{w} \bmod n$*. Then* $\left(\frac{z^2-c}{n}\right) = -1$ *and* $(z - \frac{c}{w}, n) = 1$*.*

By inspecting the proof to Proposition 3.2 in [36] we immediately obtain:

Lemma 3.3. *Suppose that* $\left(\frac{w^2-c}{n}\right) = -1$*. Then* z *is one of the values of Proposition 3.2 exactly when*

$$\frac{w^2+c}{w} \equiv \frac{z^2+c}{z} \bmod n.$$

For the z *with* $\left(\frac{z^2-c}{n}\right) = 1$ *this condition is equivalent to*

$$\frac{a'(w)}{b'(w)} \equiv \frac{a(z)}{b(z)} \bmod n.$$

Corollary 3.4. *If* $\left(\frac{w^2-c}{n}\right) = -1$ *then* $E'(w) \equiv E(z) \bmod n$ *for the* z *of Proposition 3.2 with* $\left(\frac{z^2-c}{n}\right) = 1$*. Let* $u \equiv D(E(z)) \bmod n$*. Then, according to the value of* b_2 *in the decryption process,* $u \equiv z$ *or* $\left(\frac{c}{z}\right) \bmod n$*, and, moreover, one obtains the factorization of* n *by*

$$\begin{cases} (w-u, n), & \text{if the } b_1 \text{ in the decryption process equals } 1, \\ (w-\hat{u}, n), & \text{if the } b_1 \text{ in the decryption process equals } -1. \end{cases}$$

Proof. By property (2.12) of [36] we have for all integers i

$$X_i(a_1)Y_i(a_1, b_1) \equiv X_i(a_2)Y_i(a_2, b_2) \bmod n,$$

whenever $\frac{a_1}{b_1} \equiv \frac{a_2}{b_2} \bmod n$. Therefore, for $i = e$, Lemma 3.3 implies

$$\begin{aligned} E'(w) &\equiv X_e^{-1}(\frac{w^2+c}{w^2-w})Y_e(\frac{w^2+c}{w^2-w}, \frac{2w}{w^2-c}) \equiv \\ \equiv X_e^{-1}(\frac{z^2+c}{z^2-z})Y_e(\frac{z^2+z}{z^2-z}, \frac{2z}{z^2-c}) &\equiv E(z) \bmod n. \end{aligned} \tag{8}$$

If $b_1 = 1$ then $E(z)$ is the correct encryption of z w.r.t. the forged input b_1 (relating to w) in the decryption process, since we have $\left(\frac{z^2-c}{n}\right) = 1$ which now correspond to the correct b_1 in relation to z. The decryption oracle will firstly evaluate $\sigma a(z), \sigma b(z) \bmod n$. According to Lemma 4.3 below from these one obtains, according to the value of σ, z or $\frac{c}{z} \bmod n$, and consequently a factor of n by Proposition 3.2.

If $b_1 = -1$ the oracle will evaluate $\sigma a(z), \sigma b(z) \bmod n$ as above. From these the oracle will firstly calculate $\tilde{w}$. But this already equals z or $\frac{c}{z} \bmod n$. Since $b_1 = -1$ the oracle will continue to evaluate $\frac{\tilde{w}s-c}{s-\tilde{w}}$ which it outputs as u. For obtaining the factorization obviously the last step needs to be reversed, which can be obtained by calculating $\hat{u}$, which then gives z or $\frac{c}{z} \bmod n$ as desired. □

Remark.

- Notice that congruence (8) is equivalently to

$$\alpha'(w)^{2ed} = (a'(w) + b'(w)\sqrt{c})^{2ed} \equiv \pm\alpha(z) = \pm(a(z) + b(z)\sqrt{c}) \bmod n. \tag{9}$$

- We stress that the equivalence to factoring does not rely on the choice of b_2 in the decryption routine.

3.2 Equivalence to Factoring and the CCA

The equivalence of decryption and factorization can now be demonstrated by means of ***the chosen ciphertext attack*** in the following sequence of steps.

1. Choose a message w such that $\left(\frac{w^2-c}{n}\right) = -1$.
2. Compute a cryptotext $E'(w)$ following the formulas for $b_1 = 1$ (rather than the correct $b_1 = -1$).
3. Transmit the cryptogram $C' = [E'(w), B_1, b_2]$ where $B_1 \in \{1, -1\}$ to the decrypting unit, and obtain the recovered message u.
4. Compute

$$\begin{cases} \gcd(w-u, n), & \text{if } B_1 = 1, \\ \gcd(w-\hat{u}, n), & \text{if } B_1 = -1. \end{cases}$$

   This will yield a proper factor of n.

The attack is essentially based on the fact that the parameters $a(z)$ and $b(z)$ replace the 'wrong' parameters $a'(w)$ and $b'(w)$ when $\left(\frac{w^2-c}{n}\right) = -1$. For a more detailed analysis of this we refer to Appendix A.

4 The Proposed CCA Secure Encryption Scheme

4.1 Towards Making Decryption Unique

Observe that the factorization equivalence of the Williams scheme relies on the fact of unambiguous decryption. Our goal now is to keep up this property of security, but also to enforce correct decryption. We claim that the security of the scheme only relies on the parameter b_1.

We need to be able to guarantee that no forgery on b_1 is possible. We have seen above that otherwise the system could immediately be broken by means of a CCA.

A widely employed CCA strengthening method consists in appending to the ciphertext a tag which is the basis for a **'validity check' in the decryption algorithm**. The deciphering routine will only output the cleartext if this test is passed. We will adopt this idea so as to enforce the correct b_1. The main problem now consists in developing a suitable testing condition.

It is shown in Appendix A (Corollary A.4, Lemma A.5, and the foregoing remarks) that the additional information $a(w)$, respectively $b(w)$, provably conceals w, identifies any attempted attack and also cannot be forged without knowledge of the secret key.

It turns out, however, although both $E(w)$ and $a(w)$ (or, analogously, $b(w)$) are independently secure, the combination of the two becomes insecure. *Indeed, the combination of $E(w)$ and $a(w)$ can be used to retrieve w* (cf. Proposition 6.1 below). As these attacks rely on the explicit structure of the function $a(w)$, respectively $b(w)$, we cannot include them *per se* in the ciphertext. Rather, we suggest concealing $a(w)$ and $b(w)$ by means of some **cryptographic hash function** h that maps strings of arbitrary finite length into strings of fixed length.

For the remainder of the paper, we assume that h satisfies both `preimage resistance` and `second preimage resistance`, i.e., it must be computationally infeasible to find, respectively, any input which hashes to any pre-specified output, and any second input which has the same output as any specified input. Additionally, we assume that h satisfies the additional property of `collision resistance`, i.e., it must be computationally infeasible to find two distinct inputs that hash to the same result. More detailed requirements on h will be given below.

We now show that parameter b_2 is not essential for establishing the security of the scheme. Originally the value b_2 was required in order to guarantee the correct sign when decrypting $\alpha(w), \overline{\alpha}(w)$. This is not necessary when we incorporate the hash values of $a(w)$.

Proposition 4.1. *For any $c \in \mathbb{Z}_n$ there exists exactly one $x \in \mathbb{Z}_n^*$ such that $E(x) = c$ and $b_1 = -1$. Conversely, for fixed b_1 but unknown b_2 the second step of decrypting $c = E(x)$ evaluates to $\sigma\alpha(x), \sigma\beta(x) \bmod n$, and consequently to two distinct messages $x \equiv w_1$ or $x \equiv w_{-1} \bmod n$ where w_i corresponds to the underlying $\sigma \in \{-1, 1\}$. Let $H = h(a(x))$ where h is a collision resistant hash function. Then the correct decryption x of $E(x)$ can be determined by*

$$x = \begin{cases} w_1, & \text{if } h(a(w_1)) = H, \\ w_{-1}, & \text{if } h(a(w_{-1})) = H. \end{cases}$$

Proof. The first assertion is obvious since the mapping $x \mapsto E(x)$ for a fixed b_1 induces a permutation over $\mathbb{Z}_n$. In the decryption routine, either $a(x)$ or $-a(x) \bmod n$ will be obtained. By the choice of h, only one of these equal the required hash value H. □

Corollary 4.2. *For fixed b_1 the mappings $x \mapsto E(x)$, $[c, H] \mapsto D(c)$ where H is defined in Proposition 4.1 induce a permutation of $\mathbb{Z}_n$ respectively.*

Note that the σ obtained in the decryption routine D corresponds to the parity of $a(x) \bmod n$ as input to the hash function in the encryption routine E. 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.

Lemma 4.3. *Suppose that $c = E(w)$ is being decrypted w.r.t. the wrong σ. Then the decryption obtained equals $w(-\sigma) = c\frac{1}{w} \bmod n$.*

Proof. The wrong choice of σ implies that the first step of D evaluates to $-a(w), -b(w) \bmod n$. Then, instead of $\tilde{w} = \widetilde{w}(\sigma) \equiv \frac{a(w)+1}{b(w)}$ we obtain $\widetilde{w}(-\sigma) \equiv \frac{a(w)-1}{b(w)}$. Consequently, $\widetilde{w}(\sigma)\widetilde{w}(-\sigma) \equiv \frac{a(w)^2-1}{b(w)^2} \equiv c \bmod n$. If $b_1 = 1$ then $\widetilde{w}(-\sigma) = w(-\sigma)$. For $b_1 = -1$ we obtain $w(-\sigma)$ as $\frac{\widetilde{w}(-\sigma)-c}{s-\widetilde{w}(-\sigma)} \equiv \frac{\frac{c}{\widetilde{w}(\sigma)}s-c}{s-\frac{c}{\widetilde{w}(\sigma)}} \equiv \frac{c(s-\tilde{w})}{s\tilde{w}-c} \equiv \frac{c}{w} \bmod n$. □

4.2 Further Security Requirements

Although by the introduction of the hash function, the attacks of applying Proposition 6.1 (similar to those in [5, 8, 15]) seem to be complicated, the question remains whether any attacks exploiting some *partial information* about $E(w)$, respectively $h(a(w) \bmod n)$, can successfully be mounted. This stems from the fact that any deterministic encryption function E already *per se* exposes some information on the message.

This, of course is true for any deterministic system. Moreover, also probabilistic schemes are not always secure as was vividly demonstrated by Bleichenbacher [6]. He showed that a chosen ciphertext attack on certain (probabilistic) protocols based on RSA can be carried out when only partial information about the corresponding message is leaked.

Hence, we need to ensure that no partial information on the plaintext is leaked from an object ciphertext to probabilistic polynomial time bounded attackers. Thus, we need to require **semantic security** [16] (i.e., all bits are secure based on a hard problem).

4.3 The Proposed Encryption and Decryption Schemes

In the following, 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.

If $\overline{w}$ is the message to be encrypted let $w = 0...0\overline{w}$ be the padded message of $\overline{w}$ such that $|w| = |n|$.

Encryption $\mathcal{E} = \mathcal{E}(w)$

1. Choose randomly a session key S and a randomizer R from $\{0,1\}^{|n|}$ such that for
$$w_R = w \oplus h(S \,||\, R),$$
$$S_R = S \oplus h(w_R \,||\, R),$$
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. $b_1 = -1$ as above.
3. Put $H = h\big(\underbrace{0...0\; a(w_R)}_{\text{length.}=|n|} \,||\, \underbrace{0...0\; a(S_R)}_{\text{length.}=|n|} \,||\, S\big)$.
4. Send the cryptogram $\mathcal{C} = [c_1, c_2, c_3, c_4] = [E(w_R), E(S_R), R, H]$.

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

1. Decrypt c_1 to obtain $\sigma(w_R)a(w_R) \bmod n$, $\sigma(w_R)b(w_R) \bmod n$ following the above formulas 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 above formulas for $b_1 = -1$.
3. Select $\sigma(w_R) \in \{-1, 1\}$, $\sigma(S_R) \in \{-1, 1\}$ and calculate the corresponding w_R and S_R.
4. Calculate $S = h(w_R \,||\, c_3) \oplus S_R$.
5. Check whether
$$c_4 = h\big(\underbrace{0...0\; \sigma(w_R)a(w_R)}_{\text{length.}=|n|} \,||\, \underbrace{0...0\; \sigma(S_R)a(S_R)}_{\text{length.}=|n|} \,||\, S\big). \tag{10}$$
6.
   - If step (5) returns 'true', output $w = h(S \,||\, c_3) \oplus w_R$ and S.
   - 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".

Notice that $b_1 = -1$ is fixed both in the en- and the decryption routine. Also observe that we have shown above that also the implicit b_2 will always be obtained correctly.

Lemma 4.4. *For the above routines the decryption of an encryption of any message always gives this message.*

Proof. This follows from Proposition 4.1 because the corresponding b_2 is uniquely determined by means of the hash value and b_1 is by the algorithm fixed to -1. □

4.4 Relation to Similar Concepts

In the above scheme we incorporate several existing principals of other cryptoschemes.

- Observe that the randomizer R is part of the public information because it directly appears in C. This is very similar to Canetti's [11] concept of a ' *public randomness scheme*' (cf. p. 466 of [11]) where he makes the randomizer public as well.
- The second entry in the cryptogram consists of some encryption of some random value, S. This is very similar to a ' *key encapsulation scheme*'. In [33] this defines a scheme that allows a party to generate a random bit string (in our case, S) and send it to another party, encrypted under the receiving party's public key (i.e. $E(S_R)$).

Remark. Instead of applying the encryption routine, even faster methods for realizing a key-encapsulation scheme are known. These can be used to obtain efficient and secure encryption schemes, when utilizing 'truly random' hash functions [33]. In our case we avoid the ROM by invoking one additional Williams encryption to obtain the ciphertext $\mathcal{C}$. Similarly as in [9] it seems to be difficult to avoid double encryption when excluding random oracles.

5 Definitions of Security

5.1 The Notions

To capture the task of partial security, we adopt the notion of semantic security to CCA, which we define by means of the following game-play (cf. [9]).

We distinguish between **semantic security** against (a) *adaptive chosen plaintext* attacks and semantic security against (b) *adaptive chosen ciphertext* attacks.

Let the adversary be given $A's$ public encryption key and also access to (a), the enciphering algorithm, respectively, (b) the deciphering algorithm.

In the case of (b), the adversary makes arbitrary queries to the decryption oracle, decrypting ciphertexts of his choice. (In case (a) the adversary never has access to the decryption oracle.)

Next, the adversary chooses two messages, m_0, m_1 and sends it to the encryption oracle, which chooses a bit $b \in \{0,1\}$ at random, and encrypts m_b. It then gives the resulting 'target' ciphertext C is given to the adversary. The adversary does not see the bit b. Now the challenge to the adversary consists of determining for which of the two messages C is the ciphertext.

After receiving the target ciphertext $C = C_{tar}$ from the encryption oracle, the adversary continues to query the corresponding oracle (for (a) respectively (b)), subject only to the restriction that in case (b) the query must be different to the exact object ciphertext C_{tar}.

Finally, the adversary outputs $b' \in \{0, 1\}$ which is his guess of the value b. The adversary's *advantage* is defined to be the distance from $1/2$ of the probability that $b' = b$.

The scheme is semantically secure (with respect to (a) respectively (b)) if for any efficient adversary, its advantage is negligible.

Definition 5.1. Let us call a tuple $\mathcal{C} = [c_1, c_2, c_3, c_4]$ a valid ciphertext, if the decryption oracle does not output 'NULL'. Any c_i of a valid ciphertext will be referred to as a 'valid c_i'.

Message awareness. The idea is that an adversary "knows" (can compute [2]) the decryption of the message which he encrypts in the sense that he cannot produce a ciphertext C apart from the plaintext w for which C is the ciphertext.

Observe that in our case part of the ciphertext is H which serves also for providing the 'correct' b_2. However, this H could be forged in a way to enforce the wrong σ (and thus the wrong decryption) in $\mathcal{D}$ by using $-a(x) \bmod n$ instead of the correct $a(x) \bmod n$ as the hash-input. In can easily be seen from the recursion relation of the Lucas sequences that $\frac{U_e(2a(x),1)}{b(x)V_e(2a(x),1)} \equiv \frac{U_e(-2a(x),1)}{-b(x)V_e(-2a(x),1)} \bmod n$ so that $E(x)$ w.r.t. b_2 equals $E(x)$ w.r.t. $-b_2 \bmod 2$. Therefore the encrypted values in $\mathcal{C}$ are independent of H or the underlying σ (which corresponds to the sign of $a(x)$). In order to avoid this forgery of the wrong σ, we demand the following.

Definition 5.2. We call our encryption scheme message-aware if it is infeasible for an attacker who knows the public key to produce a valid cryptogram without knowing the underlying w, S and the signs of $\pm a(w_R) \bmod n$, $\pm a(S_R)$ in the hash-input.

5.2 Requirements on the Hash Function

We say that a hashing family $\mathcal{H}$ is *suitable* if

1. $\mathcal{H}$ is preimage-resistant.

2. $\mathcal{H}$ is collision-resistant.

3. For every $h \in \mathcal{H}$ and every two distinct messages x, y, the distributions $h(x \,||\, R)$, $h(y \,||\, R)$, induced by the random choice of R, are statistically close.

Remark.

- Observe that the hashing is probabilistic: different applications of the same message result in different hash values due to the randomizer R in the input. Although in our case each block in the hash input relies on some random value (S and R), we can picture our hash functions as (special instances of) Canetti's [11] "*oracle hash*" functions. In his case one part of the input is independent of R (e.g. the message or the signature).

However, there is no compelling reason, why the other part of the input cannot be some function of R.

It is this randomization (R or S) that Canetti uses to prove that any oracle-hash function 'hides all information on its input'.

- The assumption on h imply *oracle indistinguishability* [11]: for any polynomially bounded adversary it is computationally infeasible to distinguish between hashes of x relative to the random value R and hashes of y relative to R.

6 Proof of Security

6.1 Some Properties of The Functions Involved

The following two propositions are dealing with some fundamental properties of the underlying deterministic en- and decryption schemes of section 2.2 that are being applied within our algorithms.

Proposition 6.1. *Let b_1 be fixed and $E(w)$ as well $a(w)$ be given. Then there is an efficient algorithm for evaluating the underlying message w.*

Proof. For establishing this result we adopt the ideas of the attack developed in [5]. 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

$$\phi(kx + l) \equiv k\hat{w} + l \bmod n \text{ for all } k, l \in \mathbb{Z}_n. \tag{11}$$

Since $\hat{w}^2 \equiv cB \bmod n$ it can easily be verified that ϕ is a ring homomorphism. In particular, we then have $\phi(x - s) = \hat{w} - s = \frac{c - s^2}{w + s}$, $\phi(xs - c) = w\frac{s^2 - c}{w + s}$ and consequently $\phi(\frac{xs - c}{x - s}) = w$. In R the expression $\frac{xs-c}{x-s}$ becomes $\frac{s^2-c}{s^2-cB}x + cs(B-1) \bmod n$ which we will denote by $w_1 x + w_2$. 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 $w_1 x + w_2$.

The idea behind the attack in [5, 8, 15] 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 $w_1 x + w_2$ 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 (along with the indeterminate x), 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. $\square$

Obviously, if for a valid ciphertext the decrypted message w as well as the corresponding session key S are made known then also the randomized values w_R and S_R are known.

Corollary 6.2. *Let the adversary be given any valid ciphertext and access to $\mathcal{D}$. Then in the decryption routine the correct $\sigma(w_R)$ and $\sigma(S_R)$ will be obtained, provided the adversary does not know the underlying message.*

Proof. Suppose the adversary knows the ciphertext but, prior to accessing $\mathcal{D}$, not the underlying message. In order to have the decryption routine evaluate the wrong $\sigma(w_R)$, say, H needs to correspond to the hash value where the input is $-a(w_R) \bmod n$. By the above properties on h the adversary needs to evaluate the hash function w.r.t. this input. But then he would know $-a(w_R)$ and consequently $a(w_R) \bmod n$. According to Proposition 6.1 he would know w_R. Similarly, in order to pass the test, he would need to know $E(S_R)$ along with $a(S_R)$ or $-a(S_R)$ (depending on whether he intends to forge $\sigma(S_R)$ as well). By the same argument he would then know S_R, and consequently w, a contradiction. $\square$

Corollary 6.3. *The signs of $\pm a(w_R), \pm a(S_R) \bmod n$, respectively, that pass the test in the decryption routine, respectively, are exactly the signs of the input to the hash function in the encryption routine, respectively.*

Proposition 6.4. *If there is an algorithm for retrieving $\sigma a(w) \bmod n$ from $E(w)$ (where both values correspond to the same b_1) then there is an efficient algorithm for factorizing n.*

Proof. We can assume that the algorithm finds the correct σ because otherwise the above assertion holds for $-\sigma$. Firstly, let $b_1 = -1$. Suppose that $a(w)$ can efficiently be obtained from $E(w)$. Then also the underlying w w.r.t. $b_1 = -1$ is known. Now consider $a'(w)$ and $E'(w)$ which can be calculated since w is known. We then note that $E'(w) \equiv E(z) \bmod n$ where z is the value of Proposition 3.2 (i). By assumption, from $E(z)$ the corresponding $a(z)$ (w.r.t. $b_1 = 1$) can be obtained. But according to Lemma A.3 $(a'(w) - a(z), n)$ gives the factorization of n, as claimed. $\square$

Proposition 6.5. *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. $\square$

Proposition 6.6. *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 in $\mathcal{D}$ the wrong $\sigma = \sigma(a(x))$ is obtained. By Lemma 4.3 the decryption routine evaluates $x(-\sigma) \equiv \frac{c}{x} \bmod n$. Since also $\widehat{x}(-\sigma) \equiv \frac{c}{\hat{x}} \bmod n$ the definition of a gives $a(x(-\sigma)) \equiv \frac{\left(\frac{c}{\hat{x}}\right)^2+c}{\left(\frac{c}{\hat{x}}\right)^2-c} \equiv \frac{\hat{x}^2+c}{-\hat{x}^2+c} \bmod n$ which gives the desired result. □

Corollary 6.7. *If $b_1 = -1$ and $a(x) \equiv -a(y) \bmod n$, then $\hat{x}^2 \equiv (\widehat{c/y})^2 \bmod n$.*

6.2 Security Against Chosen Plaintext Attacks

Theorem 6.8. *The above cryptosystem is semantically secure against adaptive chosen plaintext attacks assuming that the factorization of $n = pq$ is hard and h is a suitable hash function.*

Proof. Let the adversary be given some valid ciphertext (e.g. as a challenge or via eavesdropping). The proof relies on the idea that the adversary must recover the complete preimages of $E(w_R)$ and $E(S_R)$, respectively, in order to say anything meaningful about w itself. We show that if there is an adversary who is able to break the above encryption scheme then there is an algorithm that can be used to factorize the modulus n, provided h is a suitable hash function.

We firstly note that the value H in a valid cryptogram does not tell the adversary anything about $a(w_R)$ and $a(S_R)$, respectively, because the distribution of hashes w.r.t. R is equally distributed by the assumption on h. Even if the adversary retrieves either one but not both of $a(w_R)$, $a(S_R)$ it follows by the assumption on H, that this does not help him in learning anything about the other.

Consequently, the adversary's advantage is independent of the entry H in C.

Thus, the adversary's advantage merely depends on his ability of retrieving $x = w_R, S_R$ from $E(x)$. We now assume that he can do so with some probability better than guessing. But then he can apply the attack of section 3. That is, he can retrieve z from $E'(x) \equiv E(z) \bmod n$ and thus obtains the factorization of n, since $\left(\frac{w_R^2-c}{n}\right) = \left(\frac{S_R^2-c}{n}\right) = -1$ by construction. Consequently, assuming that factorization is intractable and h is suitable, an adversary cannot find the complete preimages of $E(w_R)$ and $E(s_R)$.

Well, if the adversary does not recover all of w_R from $E(w_R)$ then he will have no idea about $h(w_R||R)$, because by assumption, the distributions of $h(x||R)$ and $h(y||R)$ are statistically close. But a failure to know anything about $h(w_R||R)$ implies a failure to know anything about S, and therefore, of $w = h(S||R) \oplus w_R$ itself. Now, assuming that the adversary does recover w_R, a failure to completely recover S_R will again mean that the adversary fails to completely recover S. In the lack of complete knowledge of S, since $w = h(S||R) \oplus w_R$ is statistically close to $h(S'||R) \oplus w_R$ for any S', again the adversary can know nothing about w. □

6.3 The Proof of Knowledge

Obviously, the randomizer R, since directly occurring in the scope of C, plays a unique role. Nonetheless, we show in the sequel that this information cannot be used for any type of attack.

Lemma 6.9. *Let $\mathcal{C} = [c_1, c_2, c_3, c_4]$ be a valid ciphertext, h a suitable 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. 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 these values (e.g. by mounting a CCA). 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 test (10) passes if $c_4 = h(...||S) = h(....||S')$, where $S' = S'(c_3')$ is in the fourth step of the deciphering oracle computed as $S' = h(\sigma(w_R)w_R \mid\mid c_3') \oplus S_R$. By the choice of h necessarily $S' = S$ so that $h(\sigma(w_R)w_R \mid\mid c_3')$ has to evaluate to $S \oplus S_R$. But that would imply that the adversary has found some $c_3' \neq c_3$ that hashes to the specific output $S \oplus S_R$ and this is impossible. 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, c_3', is being selected. As above, we again need to have $S' = S$, where now $S' = h(w_R||c_3') \oplus D(c_2')$. 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, because in the validity check it is required that

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

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

By Proposition 6.6 the above identity is only possible if either $a(S_R) \equiv a(x) \bmod n$, or $a(S_R) \equiv a(c/x) \bmod n$, depending on whether the above σ's correspond or not. According to Proposition 6.5 and Corollary 6.7 this gives

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

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 decryption, x or c/x respectively. Since the decryption of c_2', as well as c_2, is being conducted w.r.t. $b_1 = -1$, the above decryption function induces by Corollary 4.2 a permutation of $\mathbb{Z}_n^*$, and the preimages, x and S_R, respectively c/x and S_R, need to be distinct. 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, it then follows that $\hat{x} \not\equiv -\widehat{S_R}$, respectively $\widehat{c/x} \not\equiv -\widehat{S_R} \bmod n$.

But then $(\hat{x} - \widehat{S_R}, n)$, respectively $(\widehat{c/x} - \widehat{S_R}, n)$ 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 6.1 asserts that the adversary can calculate x from $E(x) = c_2'$ and $a(x) \equiv a(S_R)$ respectively $-a(S_R) \bmod n$. The derived contradiction to the hypothesis of the lemma implies that the adversary cannot find 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. $b_1 = -1$). Since we have shown above that decrypting c_1' or determining this $a(y)$ is equivalent to factoring, 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 (10) 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.

The only case of possible forgery would be to calculate a forged c_4' w.r.t. a suitable hash-input. This then would it make necessary to modify each entry in $\mathcal{C}$. This case will be considered below. ☐

Theorem 6.10. *Assume that h is suitable and that it is computationally infeasible to find the factorization of n. Then the above encryption scheme is message aware.*

Proof. To prove this theorem, let us consider an adversarial algorithm that has access to h, $\mathcal{E}$, and $\mathcal{D}$. He can make t queries of the decryption oracle and then produces a new valid ciphertext that he outputs. By way of contradiction, we assume that he can do this apart from knowing the underlying message. As in [4] we demand that the adversary never outputs a string that coincides with the value returned from some $\mathcal{E}$ query (otherwise the concept of message-awareness would not make sense).

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 forged valid ciphertext that the adversary produces. By Lemma 6.9 we only need to distinguish between the following types of forgery.

- 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 $1 \le j \le t$ 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) $\mathcal{C}' = [c_1, c_2' \neq c_2, c_3, c_4']$,
  (d) $\mathcal{C}' = [c_1' \neq c_1, c_2' \neq c_2, c_3, c_4']$,
- Type III: For all $1 \le i \le t$, $\mathcal{C}' = [c_1' \neq c_1^{(i)}, c_2' \neq c_2^{(i)}, c_3' \neq c_3^{(i)}, c_4' \neq c_4^{(i)}]$.

Type I forger:

Suppose we have a type I (a) forger. 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 (10) 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 forgery under inspection.

Similarly we see that a type I (b) forgery 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) forger. 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 6.9. He needs to define the 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 (10). Similarly as in Lemma 6.9, he would be able to factorize n, a contradiction. Hence, any type I forgery will get rejected as well.

Type II forger:

For a type II (a) forger observe that by definition c_1, c_2, c_3 remain unchanged. Consequently, from the $\mathcal{C}_j$ query, he knows the underlying w, S and σ's.

Since $c_4 = h(\sigma(w_R)a(w_R)||\sigma(S_R)a(S_R)||S)$ for the specific $\sigma(w_R)$, $\sigma(S_R)$, one can only obtain a modified hash output w.r.t. different $\sigma(w_R)$ and/or $\sigma(S_R)$. In particular, the adversary then would know these σ's. Of course he also knows the underlying message w as well S that results in the given cryptogram.

In a type II (b) forgery 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 6.1, we follow that he knows 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 S and four different w. He can easily verify which of those have the desired encryptions c_1' and c_2' so that he knows the w and S that result in $\mathcal{C}'$.

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

A type II (d) forgery 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 forger:

The result follows exactly as for a type II (d) forger 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 now $c'_3 = R'$. 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 values, which completes the proof of Theorem 6.10 in all cases. □

Remark. The decryption of a valid ciphertext implies a non-interactive proof of plaintext-knowledge where the sender convinces the decrypter that he is aware of the underlying plaintext. Contrary to other existing plaintext-knowledge protocols our proof is non-interactive. In this sense, our concept is more related to the notion of plaintext awareness of [4]. We will discuss the differences in the next section.

6.4 The Proposed Concept of Message-Awareness

Observe that in the cases where the forged ciphertext do not get rejected the adversary automatically, depending on each case, knows the exact ways as *how to calculate the underlying message*. That is, by converting the above steps in the proof into corresponding algorithms, we can establish w and S.

Corollary 6.11. *Any algorithm that produces a valid cryptogram (other than by running $\mathcal{E}$) can be converted to an efficient algorithm that extracts the underlying w, S and σ's thereof.*

Remark. In [4] a similar algorithm established the basis for actually defining plaintext awareness. The idea of [4] is that the adversary who has created some cryptogram $\mathcal{C}$ can also design some auxiliary algorithm K, that eventually outputs the underlying message of this $\mathcal{C}$. Specifically, in Definition 4.1 of [4] they demand that there exists some (universal) algorithm K (the "plaintext extractor") that could output the underlying message when receiving suitable information from the adversary which is related to the ciphertext he produces. Their definition requires that K can establish the message by just looking at the adversary's oracle h and the answers to them, and the answers to his $\mathcal{E}$ calls. Note that in their case h (which we will write as $\mathtt{h}$) is modeled by some random oracle. We stress that their concept of such extractor is different from ours.

Our extractor consists also of part of the algorithm that the adversary runs to establish some forgery. More precisely, it utilizes the corresponding steps above for proving message-awareness. Our K simulates alternative ways of computing valid cryptotexts (e.g. by selecting $a(x)$ and

$E(x)$) in such a way that the underlying message can efficiently be obtained by means of some other algorithm (e.g. by applying Lemma 6.1).

Any K in the sense of [4] requires as input the adversary's `h` queries along with their answers whereas our "plaintext extractor" as input needs the information that the adversary uses for generating the forgery. Consequently, the K in [4] can only be defined in the ROM, whereas our K only requires the existence of a suitable hash function.

Moreover, Theorem 6.10 also characterizes *all* possibilities for any adversary to come up with a valid ciphertext other than by making $\mathcal{E}$ calls.

Corollary 6.12. *Any valid cryptogram that is not an $\mathcal{E}$ output, has to be the result of some type II or III forgery. In both cases, the adversary then knows the underlying w, S, as well as the underlying signs in the hash-input.*

6.5 Security Against Chosen Ciphertext Attacks

Theorem 6.10 asserts that having access to the decryption oracle does not help the adversary in gaining any new information. Therefore, a chosen ciphertext attack will not help because the adversary already "knows" the plaintext from any ciphertext whose decryption he might ask from an available decryption box.

The proof in [4] that PA implies security against CCA relies on the idea that if there is an adversary A that breaks the scheme in the CCA sense, then one can *create* an adversary A' that breaks the scheme under a plaintext attack. A' will run A and use the extractor to simulate the decryption oracle. Essentially, it will then have to simulate the production of the target ciphertext as well, which quite complicates the proof.

We now give an alternative proof.

Corollary 6.13. *The above cryptosystem is semantically secure against adaptive chosen ciphertext attacks, provided (1) h is a suitable hash function, and (2) the factorization of n is computationally infeasible.*

Proof. Suppose that the adversary does have some advantage in decrypting the target ciphertext C_{tar}. Since the challenger can pick w_b arbitrarily and since the value S is chosen randomly from its domain, the adversary cannot guess these choices that the challenger makes for generating C_{tar}. Consequently, the adversary cannot re-generate C_{tar} himself (neither by making $\mathcal{E}$ calls, nor by running any type II or III forgery, because he does not know the w_b and the S that the challenger used.) Of course, the adversary may request decryptions of cryptograms C other than C_{tar}. However, because of Theorem 6.10 he does not learn anything other than what he already knows prior to the generation of C, his advantage under a CCA is the same as under a chosen plaintext attack. Consequently the assumption implies that he would have broken the scheme by means of a chosen plaintext attack, which is a contradiction to Theorem 6.8. $\square$

Acknowledgements: I am deeply grateful to Professor H.C. Williams and Professor A. Menezes for numerous interesting discussions during my visit at the University of Waterloo, and for their many insightful and helpful suggestions. I am also thankful to the anonymous referees for their careful reading of an earlier version of this paper. Their comments and remarks have tremendously enhanced the quality of this presentation. Also, thanks are due to Professor W. B. Müller for his continued support.

References

[1] M. Bellare, P. Rogaway, Random oracles are practical: A paradigm for designing efficient protocols, Extended abstract in *Proc. First Annual Conference on Computer and Communications Security, ACM*, 1993; full paper available at `http://www.cs.ucdavis.edu/~rogaway/papers/`.

[2] M. Bellare, P. Rogaway, Optimal asymmetric encryption – How to encrypt with RSA, Extended abstract in *Advances in Cryptology - Eurocrypt 94, LNCS* 950, A. De Santis (ed.), Springer-Verlag (1995) pp. 92–111; full paper available at the above URL.

[3] M. Bellare, P. Rogaway, Minimizing the use of random oracles in authenticated encryption schemes, Shortened version in the *Proceedings of ICICS'97, LNCS* 1334, Y. Han, T. Okamoto, S. Qing (eds.), (1997) pp. 1–16.

[4] 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-Verlag (1998); full version available at `http://www-cse.ucsd.edu/users/mihir/papers/crypto-papers.html`.

[5] D. Bleichenbacher, On the Security of the KMOV Public Key Cryptosystem, *Advances in Cryptology - Crypto'97, LNCS* 1294, Springer-Verlag (1997) pp. 235 – 248.

[6] D. Bleichenbacher, Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1, *Advances in Cryptology - Crypto'98, LNCS* 1462, H. Krawczyk (ed.), Springer-Verlag (1998) pp. 1–12.

[7] D. Boneh, R. Venkatesan, Breaking RSA May Not Be Equivalent to Factoring, *Advances of Cryptology - Eurocrypt '98, LNCS* 1403, K. Nyberg (ed.), Springer-Verlag (1998) pp. 59–71.

[8] 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-Verlag (1996) pp. 1–9.

[9] 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-Verlag (1998) pp. 13–25.

[10] R. Canetti, O. Goldreich, S. Halevi, The random oracle model, revisited, In: *30 th Annual ACM Symposium on Theory of Computing* (1998).

[11] R. Canetti, Towards Realizing Random Oracles: Hash Functions That Hide All Partial Information, *Advances in Cryptology - Crypto'97*, 455-469.

[12] I. Damgård, Towards practical public key cryptosystems secure against chosen ciphertext attacks, *Advances in Cryptology - Crypto'91, LNCS* 576, J. Feigenbaum (ed.), Springer-Verlag (1992) pp. 445–456.

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

[14] R. Gennaro, S. Halevi, T. Rabin, Secure Hash-and-Sign Signatures without the Random Oracle, preprint March 22, 1999; http://philby.ucsd.edu/cryptolib/

[15] R. Gennaro, A. Shamir, Partial Cryptanalysis of Koyama's Eurocrypt'95 scheme, LCS Technical Memo 512, May 10 (1996) MIT.

[16] S. Goldwasser, S. Micali, Probabilistic Encryption, *Journal of Computer and System Sciences* 28 (April 1984) pp. 270–299.

[17] J. Håstad, M. Näslund, The Security of Individual RSA Bits, *Proceedings of FOCS '98*, IEEE.

[18] K. Kurosawa, T. Ito, M. Takeuchi, Public key cryptosystem using a reciprocal number with the same intractability as factoring a large number, *Cryptologia,* no 12 (1988) pp. 225–233.

[19] C. H. Lim, P. J. Lee, Another method for attaining security against adaptively chosen ciphertext attacks, *Advances in Cryptology - Crypto'93, LNCS* 773, D. R. Stinson (ed.), Springer-Verlag (1994) pp. 420–434.

[20] J. H. Loxton, D. D. Khoo, G. J. Bird, J. Seberry, A Cubic RSA Code Equivalent to Factorization, *Journal of Cryptology* 5 (1992) pp. 139–150.

[21] A.J. Menezes, P.C. van Oorschot, A.A. Vanstone, "Handbook of Applied Cryptography," Boca Raton, New York, London, Tokyo: CRC Press, (1997).

[22] S. Müller, W. B. Müller, The Security of Public Key Cryptosystems Based on Integer Factorization, *Information Security and Privacy – ACISP'98, LNCS* 1438, C. Boyd, E. Dawson (eds.), (1998) pp. 7–23.

[23] S. Müller, On the Security of an RSA Based Encryption Scheme. *Information Security and Privacy – ACISP'99, LNCS*, 1587, Pieprzyk, J., Safavi-Naini, R., Seberry, J. (eds.) (1999), pp. 135-148.

[24] S. Müller, Some Remarks on William's Public-Key Crypto-Functions, Manuscript 1999.

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

[26] T. Okamoto, S. Uchiyama, E. Fujisaki: EPOC: Efficient Probabilistic Public-Key Encryption, submission to P1363a (1998).

[27] C. Rackoff, D. Simon, Non-interactive zero-knowledge proofs of knowledge and chosen ciphertext attack, *Advances in Cryptology - Crypto'91, LNCS*, 576, Springer-Verlag (1991).

[28] P. Ribenboim, "The New Book of Prime Number Records", Springer Verlag, Berlin (1995).

[29] R.L. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Comm. ACM 21 (1978) pp. 120–126.

[30] R. Scheidler, A Public-Key Cryptosystem Using Purely Periodic Fields, *J. Cryptology* 11 (1998) pp. 109-124.

[31] R. Scheidler, H.C. Williams, A Public-Key Cryptosystem Utilizing Cyclotomic Fields, *Designs, Codes and Cryptography* 6 (1995) pp. 117-131.

[32] V. Shoup, Why Chosen Ciphertext Security Matters. Research Report RZ 3076, IBM Research Division Zürich (1998); `http://www.cs.wisc.edu/~shoup/papers/`.

[33] V. Shoup, Using Hash Functions as a Hedge against Chosen Ciphertext Attack, preprint June 21, 1999; http://philby.ucsd.edu/cryptolib/1999.html

[34] Y. Tsiounis, M. Yung, On the security of ElGamal-based encryption, To appear in the 1998 International Workshop on Practice and Theory in Public Key Cryptography (PKC '98); `http://www.ccs.neu.edu/home/yiannis/pubs.html`.

[35] H.C. Williams, A modification of the RSA Public-Key Encryption Procedure, *IEEE Trans. Inf. Theory*, Vol. IT-26, No. 6 (1980) pp. 726-729.

[36] H.C. Williams, Some Public-Key Crypto-Functions as Intractable as Factorization, *Cryptologia* 9 (1985) pp. 223-237.

[37] H.C. Williams, An M^3 public-Key Encryption Scheme, *Advances in Cryptology - Crypto'85, LNCS*, (1986) pp. 358 - 368.

[38] H.C. Williams, "Édouard Lucas and Primality Testing", Canadian Mathematical Society Series of Monographs and Advanced Texts, Vol. 22 (1998), John Wiley & Sons.

[39] Y. Zheng, J. Seberry, Practical approaches to attaining security against adaptively chosen ciphertext attacks, *Advances in Cryptology - Crypto'92, LNCS* 740, E.F. Brickell (ed.), Springer-Verlag (1993) pp. 292–304.

[40] Y. Zheng, J. Seberry, Immunizing public key cryptosystems against chosen ciphertext attacks, The Special Issue on Secure Communications, *IEEE Journal on Selected Areas in Communications*, Vol. 11, No. 5 (1993) pp. 715–724.

[41] Y. Zheng, Improved public key cryptosystems secure against chosen ciphertext attacks, Preprint No.94-1, 24 January (1994), Department of Computer Science, University of Wollongong; `http://www.pscit.monash.edu.au/~yuliang/pubs/`.

A Some Fundamental Properties of the Williams Scheme

Lemma A.1. *Consider the two distinct cases.*

(i) $Q \equiv \gamma(w)\overline{\gamma}(w) \bmod n$, $\alpha \equiv \frac{\gamma(w)}{\overline{\gamma}(w)} \equiv a(w) + b(w)\sqrt{c} \bmod n$,

(ii) $Q \equiv \gamma'(w)\overline{\gamma'}(w) \bmod n$, $\alpha \equiv \frac{\gamma'(w)}{\overline{\gamma'}(w)} \equiv a'(w) + b'(w)\sqrt{c} \bmod n$.

Then the following conditions hold:

- $\alpha^{(p_1 - \left(\frac{c}{p_1}\right))/2} \equiv \left(\frac{Q}{p_1}\right) \bmod p_1$ *for* $p_1 \in \{p, q\}$,
- $\left(\frac{Q}{n}\right) = \left(\frac{2(a+1)}{n}\right) = \begin{cases} 1 & \textit{when } a = a(w), b = b(w), \\ -1 & \textit{when } a = a'(w), b = b'(w). \end{cases}$

Proof. It has been shown in [36] that the first assertion holds for any $\alpha = \frac{\gamma}{\overline{\gamma}}$ where γ takes the above form irrespective of the choice of b_1, i.e. where $\gamma = \gamma(w)$ or $\gamma = \gamma'(w)$. Therefore the first assertion holds for the two cases under inspection.

For the second assertion consider firstly the case that $a = a(w), b = b(w)$. Then, by (1), $\left(\frac{Q}{n}\right) = \left(\frac{\gamma(w)\overline{\gamma}(w)}{n}\right) = \epsilon \left(\frac{w^2-c}{n}\right) = \left(\frac{2(a+1)}{n}\right)$, where $\epsilon = b_1$. Similarly, for $a = a'(w), b = b'(w)$, it can easily be seen that $\epsilon = -b_1$. □

By the definition of e and d this leads to

Corollary A.2. *Under the same conditions of Lemma A.1 we have:*

$$\begin{array}{ll} \alpha^{2ed} \equiv \pm\alpha \bmod n, & \textit{if } a = a(w), b = b(w), \\ \begin{cases} \alpha^{2ed} \equiv +\alpha \bmod p_1, \\ \alpha^{2ed} \equiv -\alpha \bmod p_2. \end{cases} & \textit{if } a = a'(w),\ b = b'(w), \end{array}$$

for $p_1, p_2 \in \{p, q\}$.

Observe that the attack heavily depends on congruence (8) which implies that the parameters $a(z)$ and $b(z)$ replace the 'wrong parameters' $a'(w)$ and $b'(w)$ where $\left(\frac{w^2-c}{n}\right) = -1$ by hypothesis.

An analogous result, however, is not true for $\left(\frac{w^2-c}{n}\right) = 1$ when utilizing false encryption with respect to $b_1' = -1$. It can easily be shown that the above ideas for establishing the factorization equivalence do not carry over to the case $\left(\frac{w^2-c}{n}\right) = 1$. The decryption of the 'false encryption' for b_1' will generally not lead to a proper factor of n. This justifies why $E'(w)$ only is defined with respect to $\left(\frac{w^2-c}{n}\right) = -1$.

Lemma A.3. *Suppose* $\left(\frac{w^2-c}{n}\right) = -1$ *and* z *is one of the values of Proposition 3.2 (i). Then, for* $p_1, p_2 \in \{p, q\}$

$$\begin{cases} a'(w) \equiv a(z) \bmod p_1, \\ a'(w) \equiv -a(z) \bmod p_2, \end{cases} \quad \textit{and} \quad \begin{cases} b'(w) \equiv -b(z) \bmod p_2, \\ b'(w) \equiv b(z) \bmod p_1. \end{cases}$$

Proof. Firstly, equation (9) asserts that $(a'(w) + b'(w)\sqrt{c})^{2ed} \equiv \pm(a(z) + b(z)\sqrt{c}) \bmod n$. Also, by Corollary A.2 we have $(a'(w)+b'(w)\sqrt{c})^{2ed} \equiv (a'(w)+b'(w)\sqrt{c}) \bmod p$, respectively $(-a'(w)-b'(w)\sqrt{c}) \bmod q$. □

Corollary A.4. *Let* w *be the message and* $[C, b_1, b_2]$ *be the input for the decrypting function* D.

- *If* $b_1 = \left(\frac{w^2-c}{n}\right)$ *and* $C = E(w)$, *then* $a(w) \equiv a(u)$, $b(w) \equiv b(u) \bmod n$, *where* $u = D(C) = D(E(w))$.
- *If* $\left(\frac{w^2-c}{n}\right) = -1$, *then* $D(E'(w)) = z$, $\gcd(a'(w) - a(z), n) = p$ *or* q, *and* $\gcd(b'(w) - b(z), n) = p$ *or* q.

Proof. The first case is obvious because $w = u$ for correct en- and decryption. For the second case observe that $a'(w)$ will be decrypted to $\pm a(z) \bmod n$ by equation (9). Therefore, $a(u) \equiv \pm a(z) \bmod n$, where z is one of the values of Proposition 3.2 (i). Finally, $a'(w) \not\equiv a(z) \bmod n$ by Lemma A.3, and analogously, $b'(w) \not\equiv b(z) \bmod n$, which gives the result. □

Thus, for $\left(\frac{w^2-c}{n}\right) = -1$, 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. Obviously, knowing either one of $a'(w)$ or $b'(w)$ suffices to identify an attack, since it is impossible for an outsider to replace $a'(w)$ or $b'(w)$ by $a(z)$ respectively $b(z)$. Moreover, we will show that w cannot be retrieved from $a(w)$ or $b(w)$ (respectively $a'(w)$ or $b'(w)$).

Lemma A.5. *Let* $\left(\frac{w^2-c}{n}\right) = -1$ *and* $\gcd(w, n) = 1$. *Then the following conditions hold.*

(i) Retrieving w *from* $a(w) \bmod n$ *is computationally equivalent to factoring* n.

(ii) Retrieving w *from* $b(w) \bmod n$ *is computationally equivalent to factoring* n.

Proof. Consider firstly case (ii) for $b_1 = 1$. Let $y = b(w) \bmod n$. Then y has the form $y \equiv \frac{2x}{x^2-c} \bmod n$. By hypothesis, the attacker can find a solution x from y. Equivalently, the attacker can find a solution of $x^2 \equiv 2\frac{x}{y} + c \bmod n$. By substituting $x' \equiv x - \frac{1}{y} \bmod n$, this equation can be rewritten as $(x')^2 \equiv x^2 - \frac{2x}{y} + \frac{1}{y^2} \equiv \left(\frac{1}{y}\right)^2 + c \bmod n$. So every solution has the form $x \equiv \sqrt{(y^{-1})^2 + c} + \frac{1}{y} \bmod n$, which gives four possible values modulo n. With probability $\frac{1}{2}$ the calculated solution will be w or $-w \bmod n$, where $w \in \mathbb{Z}_n^*$ is the original message. However, for the other two solutions x we have $\gcd(w - x, n) = p$ or q which can be evaluated by the attacker since he knows the original w. The case $b_1 = -1$, respectively case (i) can be treated in a similar way by calculating the four possible solutions modulo n of the corresponding quadratic equations. Again, exactly two of them will lead to a proper factor. □