Submission to IEEE P1363a

HD–RSA: Hybrid Dependent RSA a New Public-Key Encryption Scheme

David Pointcheval

Département d'Informatique, ENS – CNRS
45 rue d'Ulm, 75230 Paris Cedex 05, France.

E-mail: David.Pointcheval@ens.fr,
URL: http://www.di.ens.fr/~pointche.

October 1999

Abstract.

This paper describes a new hybrid RSA-based public-key encryption scheme, the HD-RSA. It relies on the recently proposed Dependent–RSA problem, which can be proven as difficult as the original RSA problem, in some circumstances. The basic scheme, using the "one-time pad" symmetric encryption, provides a both very efficient scheme and secure relative to the sole Dependent–RSA problem. A more general proposal, by integrating symmetric encryption schemes, allows much higher rates under a very weak assumption about the symmetric scheme used.

The general scheme is first presented together with a careful study of its security relative to the Dependent–RSA problem. Then, the hardness of this new problem is discussed, namely by proving its equivalence with RSA, for well-chosen exponents. Therefore, it results that this new encryption scheme is semantically secure against any kind of attacks, namely non-adaptive and even adaptive chosen-ciphertext ones.

Moreover, with a similar security as OAEP–RSA (PKCS #1 v2.0), this scheme can reach higher speed rates. Furthermore, if one compares it with the DHAES or EPOC (two other IEEE P1363a candidates for encryption), efficiency gets many times better.

Keywords: Public-Key Encryption, Hybrid Scheme, Semantic Security, Chosen-Ciphertext Attacks, Integer Factoring, the Dependent–RSA Problem

Table of Contents

1 Preliminaries

This paper proposes a new hybrid method for encrypting messages using both the Dependent–RSA problem and any one-time secure symmetric encryption scheme. It is freely derived from the Eurocrypt '99 paper [23], which is added in appendix. Let us first more formally define the required background.

1.1 The Dependent–RSA Problems

For all the problems presented below, we are given a large composite RSA modulus $N = pq$ and an exponent e relatively prime to $\varphi(N)$, the totient function of the modulus N. Let us define a first new problem called the *Computational Dependent–RSA Problem* ($\mathsf{C\text{–}DRSA}$).

Definition 1 (The Computational Dependent–RSA: $\mathsf{C\text{–}DRSA}(N, e)$).

Given: $\alpha \in \mathbb{Z}_N^\star$;

Find: $(a+1)^e \bmod N$, where $\alpha = a^e \bmod N$.

Notation: We denote by $\mathsf{Succ}(\mathcal{A})$ the success probability of an adversary $\mathcal{A}$ in finding $(a+1)^e \bmod N$: $\mathsf{Succ}(\mathcal{A}) = \Pr_a[\mathcal{A}(a^e \bmod N) = (a+1)^e \bmod N]$.

As it has already been done with the Diffie-Hellman problem [14, 8], we can define a decisional version of this problem, which we therefore call the *Decisional Dependent–RSA Problem* ($\mathsf{D\text{–}DRSA}$): Given a candidate to the Computational Dependent–RSA problem, is it the right solution?

Definition 2 (The Decisional Dependent–RSA: $\mathsf{D\text{–}DRSA}(N, e)$).

Problem: Distinguish the two distributions

$$\mathcal{R}and = \left\{ (\alpha, \gamma) = (a^e \bmod N, c^e \bmod N) \,\middle|\, a, c \stackrel{R}{\leftarrow} \mathbb{Z}_N^\star \right\},$$
$$\mathcal{DRSA} = \left\{ (\alpha, \gamma) = (a^e \bmod N, (a+1)^e \bmod N) \,\middle|\, a \stackrel{R}{\leftarrow} \mathbb{Z}_N^\star \right\}.$$

Notation: We denote by $\mathsf{Adv}(\mathcal{A})$ the advantage of a distinguisher $\mathcal{A}$ in distinguishing both distributions $\mathcal{R}and$ and $\mathcal{DRSA}$:

$$\mathsf{Adv}(\mathcal{A}) = \left| \Pr_{\mathcal{R}and}[\mathcal{A}(\alpha, \gamma) = 1] - \Pr_{\mathcal{DRSA}}[\mathcal{A}(\alpha, \gamma) = 1] \right|.$$

1.2 One-Time Secure Symmetric Encryption Schemes

A *symmetric encryption scheme* allows users which share a common key to achieve confidentiality. It consists of two inverse algorithms $(\mathcal{E}^{\mathsf{sym}}, \mathcal{D}^{\mathsf{sym}})$,

- the encryption algorithm $\mathcal{E}^{\mathsf{sym}}$ takes as input a secret key sk together with a message m and outputs the ciphertext $c = \mathcal{E}^{\mathsf{sym}}_{\mathsf{sk}}(m)$,
- the decryption algorithm $\mathcal{D}^{\mathsf{sym}}$ takes as input a secret key sk together with a ciphertext c and outputs the plaintext $m = \mathcal{D}^{\mathsf{sym}}_{\mathsf{sk}}(c)$.

They should both satisfy $\mathcal{D}^{\mathsf{sym}}_{\mathsf{sk}}(\mathcal{E}^{\mathsf{sym}}_{\mathsf{sk}}(m)) = m$.

Such a symmetric encryption scheme is said *one-time secure* if for any adversary $\mathcal{A}$, after having chosen two messages m_0 and m_1 of same length, and got the encryption c of one of them, the adversary can not guess which one has been encrypted, with probability significantly greater than one half. More formally, we can give the following definition.

Definition 3 (One-Time Security). A symmetric encryption scheme is said (ℓ, t, ε)-*one-time secure* if for any adversary $\mathcal{A} = (A_1, A_2)$ with running time bounded by t,

$$\mathsf{Adv}^{\mathsf{ots}}(\mathcal{A}) \stackrel{\mathsf{def}}{=} 2 \times \Pr\left[\begin{array}{l}(m_0, m_1, s) \leftarrow A_1(1^\ell) \\ \mathsf{sk} \stackrel{R}{\leftarrow} SK, b \stackrel{R}{\leftarrow} \{0,1\} \\ c \leftarrow \mathcal{E}^{\mathsf{sym}}_{\mathsf{sk}}(m_b)\end{array} : A_2(s, c) = b\right] - 1 \leq \varepsilon,$$

with the restriction that messages m_0 and m_1 output by A_1 are both ℓ-bit strings.

The best example of such *one-time secure* symmetric encryption scheme is the One-Time Pad:

$$\mathcal{E}^{\mathsf{sym}}_{\mathsf{sk}}(m) = \mathsf{sk} \oplus m \qquad\qquad \mathcal{D}^{\mathsf{sym}}_{\mathsf{sk}}(c) = \mathsf{sk} \oplus c.$$

Theorem 1. *If the key space SK is $\{0,1\}^k$, then for any time t, the One-Time Pad encryption is $(k, t, 0)$-one-time secure: it is a* ***perfectly one-time secure*** *encryption scheme.*

To link the *one-time security* notion with more classical ones, it is equivalent to the basic semantic security, which is briefly recalled below, but using no plaintext nor ciphertext attacks. It is therefore a very weak assumption for non-perfect schemes. Indeed, for many schemes, and namely the recent AES candidates, cryptanalysts fail using *adaptive* chosen–plaintext/ciphertext attacks.

1.3 Chosen-Ciphertext Security

In 1984, Goldwasser and Micali [18] defined some security notions that any encryption scheme should satisfy (symmetric or asymmetric), namely *indistinguishability of encryptions* (a.k.a. *polynomial security* or *semantic security*). This notion means that a ciphertext does not leak any useful information about the plaintext, but its length, to a polynomial time attacker. For example, if an attacker knows that the plaintext is either "sell" or "buy", the ciphertext does not help her.

More formally, an attacker is seen as a two-stage ("find-and-guess") Turing machine which first chooses two messages, during the "find"-stage. In the second stage, the "guess"-stage, she receives a challenge, which is the encryption of one of both chosen messages, and has to guess which one is the corresponding plaintext.

During the last ten years, beyond semantic security, a new security notion has been defined: the *non-malleability* [15, 5]. Moreover, some stronger scenarios of attacks have been considered: the *(adaptive) chosen-ciphertext attacks* [20, 25]. More precisely, the non-malleability property means that any attacker cannot modify a ciphertext while keeping any control over the relation between the resulting plaintext and the original one.

In the public-key setting, any attacker can play a *chosen-plaintext attack*, since she can encrypt any message she wants. However, stronger attacks has been defined. First, Naor and Yung [20] defined the *chosen-ciphertext attack* (a.k.a. *lunchtime attack*) where the attacker has access to a decryption oracle during the "find"-stage, to choose the two plaintexts. Then, Rackoff and Simon [25] improved this notion, giving the decryption oracle access to the attacker in both stages (with the trivial restriction not to ask the challenge ciphertext). This attack is known as *adaptive chosen-ciphertext attack* and is the strongest that an attacker can play, in the classical model.

Few years later, another kind of property for encryption schemes has also been defined, called *Plaintext-Awareness* [4], which means that no one can produce a valid ciphertext without knowing the corresponding plaintext.

At Crypto '98, Bellare *et al.* [2] provided a precise analysis of all these security notions. The main practical result is the equivalence between non-malleability and semantic security in adaptive chosen-ciphertext scenarios, therefore called the *Chosen-Ciphertext Security*.

1.4 The Random Oracle Model

The best security argument for a cryptographic protocol is a reduction from a well-studied difficult problem, in the sense of the complexity theory. The classically referred problems are RSA [26], the factorization or the discrete logarithm. But no really efficient cryptosystem can aspire to such a strong argument. Indeed, the best encryption scheme that achieves chosen-ciphertext security in this sense was proposed at Crypto '98 by Cramer and Shoup [13], and still requires more than four exponentiations for an encryption. Furthermore, it relies on the weakest problem known as the Decisional Diffie–Hellman problem [8], which requires particular settings to be difficult.

In 1993, Bellare and Rogaway [3] defined a model, the so-called "Random Oracle Model", where some objects are idealized, namely hash functions which are assumed perfectly random. This helped them to design later OAEP [4], the most efficient encryption scheme known until now. In spite of a recent paper [9] making people to be careful with the random oracle model, the security of OAEP has been widely agreed and became the new RSA encryption standard PKCS #1 v2.0 [27].

Moreover, one can also remark that other IEEE P1363a candidates, namely EPOC [21] and PSEC [22], are also proven secure in the random oracle model. About, DHAES [1], the adaptive HDH independence assumption is quite non-standard, and is somewhat similar to the random oracle model.

Furthermore, an important feature of the random oracle model is to provide efficient reductions between a well-studied mathematical problem and an attack. Therefore, the reduction validates protocols together with practical parameters. Whereas huge-polynomial reductions, which can hardly be avoided in the standard model, only prove asymptotic security, for large parameters. As a conclusion, it is better to get an efficient reduction in the random oracle model than a complex reduction in the standard model, since this latter does not prove anything for practical sizes and therefore actual implementations.

1.5 Related Work

For two years, many standardized protocols have been maltreated by cryptanalysts, namely PKCS #1 v1.5 [27, 6] and ISO 9796-1 [12, 11]. Indeed, for a long time, the security of practical schemes was only heuristic.

Because of that, people became aware of the necessity of provably security. As said above, the random oracle model is very well suited to provide security arguments for practical schemes. Concerning asymmetric encryption, RSA-OAEP [4] was the first example. It is a generic conversion from any trapdoor one-way permutation into a chosen-ciphertext secure encryption scheme. The only interesting application is RSA. It has been adopted by *RSA Data Security* has the new version for PKCS#1. More recently, Fujisaki and Okamoto [16, 17] and Pointcheval [24] have presented other generic conversions from any weakly secure encryption scheme into chosen-ciphertext secure ones. However the efficiency of the resulting scheme is not optimal. This drawback can be seen in the EPOC proposal [21]: the decryption phase requires a new encryption. However those schemes are the only encryption schemes provably secure relative to factorization or RSA. The present work describes the main scheme among those proposed in the paper published in Eurocrypt '99 [23], which are all provably secure relative to the Dependent–RSA problem.

2 Description of the Hybrid Dependent RSA Scheme

In this section, we first describe the general Hybrid Dependent RSA cryptosystem, with any *one-time secure* symmetric encryption scheme. Then, we present the particular case with the one-time pad. They are both recalled on Figures 1 and 2.

2.1 The Hybrid Dependent RSA Cryptosystem

For a given security parameter k, one derives the three following sub-parameters k_1, k_2 and k_3. Let us now present the new cryptosystem, where g and h are some given hash functions which output k_1-bit numbers and k_2-bit numbers respectively. It furthermore uses a symmetric encryption $(\mathcal{E}^{\mathsf{sym}}, \mathcal{D}^{\mathsf{sym}})$ with k_1-bit long keys, which can encrypt securely ℓ-bit messages.

- Key generation algorithm $\mathcal{K}(1^k)$: it generates two $k_3/2$-bit long primes p and q and computes the RSA modulus $N = pq$. It randomly chooses an exponent e, relatively prime to $\varphi(N) = (p-1)(q-1)$, and computes $d = e^{-1} \bmod \varphi(N)$.
  The public key is the pair (N, e), whereas the secrete key is the exponent d.

- Encryption algorithm $\mathcal{E}_{(N,e)}(m)$: it randomly chooses $r \in_R \mathbb{Z}_N^\star$ and computes $A = r^e \bmod N$ as well as $B = (r+1)^e \bmod N$. It then gets the secret session key $K = g(B)$ and computes $C = \mathcal{E}_K^{\mathsf{sym}}(m)$ and $H = h(m, K, r)$, which completes the ciphertext: $c = (A, C, H)$.

- Decryption algorithm $\mathcal{D}_d(c)$: from $c = (A, C, H)$, it first extracts r, by computing $A^d \bmod N$, and then gets $B = (r+1)^e \bmod N$. It can therefore recover the session key $K = g(B)$ as well as the message $m = \mathcal{D}_K^{\mathsf{sym}}(C)$ which is output only if $H = h(m, K, r)$. Otherwise, it outputs "Reject".

2.2 The OTP–Dependent RSA Cryptosystem

If one uses the "One-Time Pad" as one-time secure encryption scheme, it gets the following particular scheme, with a better security result at the price of a slightly worse efficiency.

Setting

k, security parameter, from which one derives

- k_1, size of the *one-time secure* symmetric encryption secret key ($SK = \{0,1\}^{k_1}$)
- k_2, error parameter
- k_3, size of the RSA modulus

Two hash functions: $g : \{0,1\}^{k_3} \to \{0,1\}^{k_1}$ and $h : \{0,1\}^{\ell} \times \{0,1\}^{k_1} \times \{0,1\}^{k_3} \to \{0,1\}^{k_2}$

Key Generation

$N = pq$, a large RSA modulus, with two $k_3/2$-bit long primes

e, an exponent, relatively prime to $\varphi(N)$, d its inverse modulo $\varphi(N)$

Public key: (N, e) **Secret key:** d

Encryption of $m \in \{0,1\}^{\ell}$

$r \in_R \mathbb{Z}_N^\star$, $A = r^e \bmod N$ and $B = (r+1)^e \bmod N$

$K = g(B)$ and $C = \mathcal{E}_K^{\mathsf{sym}}(m)$

$H = h(m, K, r)$, then, $c = (A, C, H)$

Decryption of $c = (A, C, H)$

$r = A^d \bmod N$ and $B = (r+1)^e \bmod N$

$K = g(B)$ and $m = \mathcal{D}_K^{\mathsf{sym}}(C)$

if $H = h(m, K, r)$ then return m, otherwise return “Reject”

Fig. 1: Hybrid Dependent RSA

Setting

k, security parameter, from which one derives

- k_1, size of the plaintext
- k_2, error parameter
- k_3, size of the RSA modulus

Two hash functions: $g : \{0,1\}^{k_3} \to \{0,1\}^{k_1}$ and $h : \{0,1\}^{k_1} \times \{0,1\}^{k_1} \times \{0,1\}^{k_3} \to \{0,1\}^{k_2}$

Key Generation

$N = pq$, a large RSA modulus, with two $k_3/2$-bit long primes

e, an exponent, relatively prime to $\varphi(N)$, d its inverse modulo $\varphi(N)$

Public key: (N, e) **Secret key:** d

Encryption of $m \in \{0,1\}^{k_1}$

$r \in_R \mathbb{Z}_N^\star$, $A = r^e \bmod N$ and $B = (r+1)^e \bmod N$

$K = g(B)$ and $C = m \oplus K$

$H = h(m, K, r)$, then, $c = (A, C, H)$

Decryption of $c = (A, C, H)$

$r = A^d \bmod N$ and $B = (r+1)^e \bmod N$

$K = g(B)$ and $m = C \oplus K$

if $H = h(m, K, r)$ then return m, otherwise return “Reject”

Fig. 2: OTP–Dependent RSA

We still use g and h, which are some given hash functions that output k_1-bit numbers and k_2-bit numbers respectively.

- Key generation algorithm $\mathcal{K}(1^k)$: it generates two large primes p and q and computes the RSA modulus $N = pq$. It randomly chooses an exponent e, relatively prime to $\varphi(N) = (p-1)(q-1)$, and computes $d = e^{-1} \bmod \varphi(N)$.
  The public key is the pair (N, e), whereas the secrete key is the exponent d.

- Encryption algorithm $\mathcal{E}_{(N,e)}(m)$: it randomly chooses $r \in_R \mathbb{Z}_N^\star$ and computes $A = r^e \bmod N$ as well as $B = (r+1)^e \bmod N$. It then gets the secret session key $K = g(B)$ and computes $C = K \oplus m$ and $H = h(m, K, r)$, which completes the ciphertext: $c = (A, C, H)$.

- Decryption algorithm $\mathcal{D}_d(c)$: from $c = (A, C, H)$, it first extracts r, by computing $A^d \bmod N$, and then gets $B = (r+1)^e \bmod N$. It can therefore recover the session key $K = g(B)$ as well as the message $m = C \oplus K$ which is output only if $H = h(m, K, r)$. Otherwise, it outputs "Reject".

This specific scheme can only encrypt k_1-bit long messages, whereas the general one can encrypt larger messages depending on the "one-time secure" symmetric encryption scheme and the required security. Furthermore, in the OTP–Dependent RSA, one can just commit the pair (m, r) in H, as done in the original paper [23], because of the bijectivity, for any message m, of $\mathsf{sk} \mapsto \mathcal{E}^{\mathsf{sym}}_{\mathsf{sk}}(m)$ when one uses the one-time pad. Anyway, we keep the K to stick to the general description.

3 Security Assessment

3.1 The Dependent–RSA Problems and RSA

In order to study those Dependent–RSA problems, we define a new one, we call the *Extraction Dependent–RSA Problem* (E–DRSA):

Given: $\alpha = a^e \in \mathbb{Z}_N^\star$ and $\gamma = (a+1)^e \in \mathbb{Z}_N^\star$;

Find: $a \bmod N$.

One can then prove that extraction of e-th roots is easier than the Computational Dependent–RSA problem and the Extraction Dependent–RSA problem together.

Theorem 1. $\mathbf{RSA}(\mathbf{N}, \mathbf{e}) \Longleftrightarrow$ *E–DRSA*$(\mathbf{N}, \mathbf{e}) +$ *C–DRSA*$(\mathbf{N}, \mathbf{e})$.

Proof. Let us be given two adversaries: $\mathcal{A}$ an E–DRSA adversary and $\mathcal{B}$ a C–DRSA adversary. For a given $c = a^e \bmod N$, an element of $\mathbb{Z}_N^\star$, whose e-th root is wanted, one uses $\mathcal{B}$ to obtain $(a+1)^e \bmod N$ and gets a from $\mathcal{A}(a^e \bmod N, (a+1)^e \bmod N)$.

The opposite direction is trivial, since extraction of e-th roots helps to solve all the Dependent–RSA problems. □

Furthermore, it is clear that any decisional problem is easier to solve than its related computational version, and trying to extract a, it is easy to decide whether the given γ is the right one. Finally, for any (N, e), the global picture is

$$\mathsf{C\text{–}DRSA} + \mathsf{E\text{–}DRSA} \Longleftrightarrow \mathbf{RSA} \Longrightarrow \mathsf{C\text{–}DRSA}, \mathsf{E\text{–}DRSA} \Longrightarrow \mathsf{D\text{–}DRSA},$$

where $A \Longrightarrow B$ means that an oracle that breaks A can be used to break B within a time polynomial in the bit-size of N, and mostly constant.

3.2 Attacks against the Extraction Dependent–RSA Problem

In order to use these problems in cryptography, we need to know their practical difficulty, for reasonable sizes. Hopefully, some of them have already been studied in the past. Indeed, they are related to many properties of the RSA cryptosystem, namely its malleability, its security against related-message attacks [10] and in the multicast setting [19].

Concerning the Extraction Dependent–RSA problem, some methods have been proposed by Coppersmith *et al.* [10], trying to solve the related-message system:

$$\begin{cases} \alpha = m^e \bmod N \\ \beta = (m+1)^e \bmod N \end{cases}$$

The most efficient method comes from the remark that m is a root for both the polynomials P and Q over the ring $\mathbb{Z}_N$, where

$$P(X) = X^e - \alpha \text{ and } Q(X) = (X+1)^e - \beta.$$

Then $X - m$ is a divisor of the gcd of P and Q. Furthermore, one can see that with high probability, it is exactly the gcd. A straightforward implementation of Euclid's algorithm takes $\mathcal{O}(e^2)$ operations in the ring $\mathbb{Z}_N$. More sophisticated techniques can be used to compute the gcd in $\mathcal{O}(e \log^2 e)$ time [28]. Then, this second method fails as soon as e is greater than 2^{60}. More formally, this remark leads to the following theorem, where $|\cdot|$ denotes the bit-length.

Theorem 2. *There exist algorithms that solve the problem* $\mathsf{E\text{–}DRSA}(N, e)$ *in linear time in both* $|N|^2$ *and* $e \times |e|^2$*.*

3.3 Consequences on the Computational Dependent–RSA Problem

Since the RSA cryptosystem appeared [26], many people have attempted to find weaknesses. Concerning the malleability of the encryption, the multiplicative property is well-known. In other words, it is easy to derive the encryption of $m \times m'$ from the encryption of m, for any m', without knowing the message m itself. However, from the encryption of an unknown message m, nothing has been found to derive the encryption of $m+1$ whatever the exponent e may be. Furthermore, using both the Theorem 1 and the Theorem 2, one can claim the following result.

Theorem 3. *There exists a reduction from the RSA problem to the Computational Dependent–RSA problem in linear time in both* $|N|^2$ *and* $e \times |e|^2$*.*

Then, for any fixed exponent e, $RSA(N,e)$ is reducible to $\mathsf{C\text{--}DRSA}(N,e)$ polynomially in the size of N, since the Extraction Dependent–RSA problem is "easy" to solve, using the gcd technique (as remarked above). Anyway, computation of e-th roots seems always required to solve the Computational Dependent–RSA problem, which is intractable for any exponent e, according to the RSA assumption.

Conjecture 1. The Computational Dependent–RSA problem is intractable for RSA moduli large enough.

Remark 1. Because of the Theorem 3, this conjecture holds for small exponents, under the RSA assumption.

3.4 About the Decisional Dependent–RSA Intractability

The gcd technique seems to be the best known attack against the Decisional Dependent–RSA problem and is impractical as soon as the exponent e is greater than 2^{60}. Which leads to the following conjecture:

Conjecture 2. The Decisional Dependent–RSA problem is intractable as soon as the exponent e is greater than 2^{60}, for RSA moduli large enough.

However, even if this second conjecture does not hold, all the following remains true, since it just relies on the Computational Dependent–RSA problem, and therefore just on the first conjecture.

3.5 Chosen-Ciphertext Security Results

Theorem 4. *The Hybrid Dependent RSA encryption scheme is semantically secure, even against adaptive chosen-ciphertext attacks, relative to the Dependent–RSA problem and the* one-time security *of the symmetric encryption scheme, in the random oracle model.*

More precisely, one can claim the following exact security result.

Theorem 5. *An adversary $\mathcal{A}$ against the semantic security of the Hybrid Dependent RSA encryption scheme, between ℓ-bit long messages, within a time bounded by t, with advantage ε, after q_D, q_G and q_H queries to the decryption oracle, and the hash functions g and h respectively, can break, for any $\nu < \varepsilon$,*

- *either the Computational Dependent–RSA problem with probability greater than*

$$\frac{1}{q_G} \times \left(\frac{\varepsilon - \nu}{2} - \frac{q_D}{2^{k_2}} \right)$$

  within a time bound t.

- *or the* one-time security*, between ℓ-bit long messages, of the symmetric encryption scheme with advantage greater than ν within a time bound t.*

Proof. For proving this result, let us assume that the symmetric encryption scheme is (ℓ, t, ν)-one-time secure for some $\nu < \varepsilon$. The semantic security of the Hybrid Dependent RSA scheme comes from the fact that any attacker cannot gain any further advantage in distinguishing the original plaintext if she has not asked for any $(\star, \star, r)$ to h (which is called "event 1" and denoted by E_1) or for $B = (r+1)^e \bmod N$ to g (which is called "event 2" and denoted by E_2).

Therefore, for a given $\alpha = a^e \bmod N$, either we learn the e-th root a of α, or the solution $(a+1)^e \bmod N$ is in the list of the queries asked to g. Both cases lead to the computation of $(a+1)^e \bmod N$.

More precisely, let $\mathcal{A} = (A_1, A_2)$ be an adversary against the semantic security of the Hybrid Dependent RSA encryption scheme, using an adaptive chosen-ciphertext attack. Within a time bound t, she asks q_D queries to the decryption oracle and q_G and q_H queries to the hash functions g and h respectively, and distinguishes the right plaintext with an advantage greater than ε. We then use her to provide an algorithm that solves the Computational Dependent–RSA problem, simply filtering the queries asked to the hash function g.

Actually, in the random oracle model, because of the randomness of g and h, if no critical queries have been asked, the attacker gets $C = \mathcal{E}^{\mathsf{sym}}_{\mathsf{sk}}(m_b)$ for a randomly chosen secret key sk and then cannot gain any advantage greater than ν, since its running time is bounded by t, and the messages are ℓ-bit long. Then,

$$\Pr_b[A_2(s, m_0, m_1, \mathcal{E}(m_b)) = b \mid \neg(\mathsf{E}_1 \vee \mathsf{E}_2)] \le \frac{1}{2} + \frac{\nu}{2}.$$

And therefore,

$$\begin{aligned}
&\Pr_b[A_2(s, m_0, m_1, \mathcal{E}(m_b)) = b] = \frac{1}{2} + \frac{\varepsilon}{2} \\
&= \Pr_b[A_2 = b \wedge \neg(\mathsf{E}_1 \vee \mathsf{E}_2)] + \Pr_b[A_2 = b \wedge (\mathsf{E}_1 \vee \mathsf{E}_2)] \\
&= \Pr_b[A_2 = b \mid \neg(\mathsf{E}_1 \vee \mathsf{E}_2)] \times \Pr_b[\neg(\mathsf{E}_1 \vee \mathsf{E}_2)] + \Pr_b[A_2 = b \wedge (\mathsf{E}_1 \vee \mathsf{E}_2)] \\
&\le \left(\frac{1}{2} + \frac{\nu}{2}\right) + \Pr_b[\mathsf{E}_1 \vee \mathsf{E}_2].
\end{aligned}$$

This leads to $\Pr[\mathsf{E}_1 \vee \mathsf{E}_2] \ge (\varepsilon - \nu)/2$.

Since we are in an adaptive chosen-ciphertext scenario, we have to simulate the decryption oracle, or to provide a plaintext-extractor, which would prove the *plaintext-awareness* of the scheme, and therefore the *chosen-ciphertext security* [4, 2].

When the adversary asks a query (A', C', H'), the simulator looks in the table of the queries previously made to the hash function h which led to H'. If H' has never been returned by h, then the simulator returns the reject symbol "$\star$". Otherwise, for any triple (m, K, r) from which h has answered H', the simulator checks whether $A' = r^e \bmod N$, if $B = (r+1)^e \bmod N$ has been asked to g and answered by K and whether $C' = \mathcal{E}^{\mathsf{sym}}_K(m)$. Then it returns m as the decryption of the triple (A', C', H'). Otherwise, the simulator considers that it is an invalid ciphertext and returns the reject symbol "$\star$".

Some decryptions may be incorrect, but only refusing a valid ciphertext: a ciphertext is refused if the query (m, K, r) has not been asked to h, or B not asked to g (or did not lead to K). However, the attacker might have guessed the right value for $h(m, K, r)$, but only with probability $1/2^{k_2}$. Using this plaintext-extractor, we obtain,

$$\Pr[(\mathsf{E}_1 \vee \mathsf{E}_2) \wedge \text{ no incorrect decryption}] \geq \frac{\varepsilon - \nu}{2} - \frac{q_D}{2^{k_2}}.$$

For the reduction, one just has to randomly choose a g-query which should correspond to $(a+1)^e \bmod N$. With probability greater than $1/q_G$, it is a good choice (or maybe, event 1 happens, but we assume the worst case). Then, with probability greater than

$$\frac{1}{q_G} \times \left(\frac{\varepsilon - \nu}{2} - \frac{q_D}{2^{k_2}}\right)$$

within roughly the same running time as the adversary $\mathcal{A}$, one obtains the right value for $(a+1)^e \bmod N$ corresponding to the given $\alpha = a^e \bmod N$. □

In particular, one can claim the following result about the OTP–Dependent RSA scheme.

Corollary 1. *Let us consider any adversary $\mathcal{A}$ against the semantic security of the OTP–Dependent RSA encryption scheme. If we assume that within a running time bounded by t it can gain an advantage ε, after q_D, q_G and q_H queries to the decryption oracle, and the hash functions g and h respectively, then one can break the Computational Dependent–RSA problem with probability greater than*

$$\frac{1}{q_G} \times \left(\frac{\varepsilon}{2} - \frac{q_D}{2^{k_2}}\right),$$

within the same running time t.

3.6 Application to Small Exponents

What we have shown is that the security of this scheme relies on the Computational Dependent–RSA problem, then it is even secure with small exponents (and even more secure with small exponents, since it is therefore equivalent to RSA).

Let us now assume we use a small exponent e, then among all the queries asked to g, one can check, easily breaking the Extraction Dependent–RSA problem, which one is the good one. Otherwise, it returns "Fail".

Theorem 6. *An adversary $\mathcal{A}$ against the semantic security of the Hybrid Dependent RSA encryption scheme with a small exponent, between ℓ-bit long messages, within a time bounded by t, with advantage ε, after q_D, q_G and q_H queries to the decryption oracle, and the hash functions g and h respectively, can break, for any $\nu < \varepsilon$,*

- *either the RSA problem with probability greater than $(\varepsilon - \nu)/2 - q_D/2^{k_2}$ within a time bound close to t.*
- *or the* one-time security*, between ℓ-bit long messages, of the symmetric encryption scheme with advantage greater than ν within a time bound t.*

About the OTP–Dependent RSA version with a small exponent, one gets the following corollary.

Corollary 2. *Let us consider any adversary $\mathcal{A}$ against the semantic security of the OTP–Dependent RSA encryption scheme with a small exponent. If we assume that within a running time bounded by t it can gain an advantage ε, after q_D, q_G and q_H queries to the decryption oracle, and the hash functions g and h respectively, then one can break the RSA problem with probability greater than $\varepsilon/2 - q_D/2^{k_2}$ within a time bound close to t.*

4 Advantages of these Schemes

4.1 Security

First, one can remark that the OTP–Dependent RSA scheme only relies on the RSA assumption, if one uses small exponents, in the random oracle model, whereas DHAES [1] furthermore requires a secure MAC.

If one wants to compare it with EPOC [21], we can claim for both a similar security level: indeed, the RSA assumption is somewhat similar to factorization. Moreover, both reductions are linear, which means that concrete security is optimal in both cases: an adversary which is able to break semantic security against chosen-ciphertext attacks with advantage ε within time t can be used to break the underlying problem with probability $\varepsilon/2$ within time very close to t.

4.2 Efficiency

Precomputations. First, one has to remark that, in the same vein as a Eurocrypt '98 paper [7], our scheme allows precomputations. Indeed, a user can precompute many triples for a given recipient, *i.e.*, $(r, A = r^e \bmod N, K = g((r+1)^e \bmod N))$. Then an encryption only requires a symmetric encryption and a hashing. However, to be fair, in the following, we won't consider this feature. Furthermore, we will just compare efficiency of the OTP–Dependent RSA scheme with other RSA/factorization based schemes: RSA–OAEP [4] and EPOC-2 [21].

RSA–OAEP. An authority chooses and publishes two hash functions g and h which both output n-bit strings. Each user chooses a large RSA modulus $N = pq$ of size $2n$ together with an exponent e. He publishes both and keeps secret the private exponent $d = e^{-1} \bmod \varphi(N)$.

To encrypt a message $m \in \{0,1\}^{n-k_1}$, one has to choose a random element $r \in \{0,1\}^n$, computes $A = (m\|0^{k_1}) \oplus g(r)$ and $B = r \oplus h(A)$ and finally sends $C = (A\|B)^e \bmod N$. The recipient can recover the message from C first computing $A\|B = C^d \bmod N$, then $r = B \oplus h(A)$ and $M = A \oplus g(r)$. If M ends with k_1 zero bits, then m is the beginning of M.

This encryption scheme essentially requires one exponentiation to the power e per encryption and one exponentiation to the power d, using Chinese Remaindering Theorem, per decryption.

EPOC-2. This scheme is also an hybrid one, but we just consider the one-time pad encryption scheme.

An authority chooses and publishes two hash functions G and H which both output n-bit strings. Each user chooses two large primes p and q of size n and publishes $N = p^2 q$. He also chooses $g \in \mathbb{Z}_N^\star$ such that the order of $g_p = g^{p-1} \bmod p^2$ is p, as well as an independent element $h_0 \in \mathbb{Z}_N^\star$, and computes $h = h_0^N \bmod N$.

To encrypt a message $m \in \{0,1\}^n$, one has to choose a random element $r \in \{0,1\}^n$ and computes $s = H(m, r)$ as well as $A = g^r h^s \bmod N$. Then one gets $B = m \oplus G(r)$. The ciphertext C consists of the pair (A, B). The recipient can first recover the random r from $L(A^{p-1})/L(g_p) \bmod p$, where $L(x) = (x-1)/p$ for any x such that $x = 1 \bmod p$. Then he gets $M = B \oplus G(r)$. M is considered as the plaintext if $A = g^r h^{H(M,r)} \bmod N$.

This encryption scheme essentially requires two exponentiation to some n-bit powers per encryption and a little more per decryption, since the receiver has to re-encrypt to verify the validity of the ciphertext. But then, he can use the Chinese Remaindering Theorem.

Efficiency Comparison. One can see, on Figure 3, a brief comparison table involving our OTP–Dependent RSA scheme together with the RSA cryptosystem and its OAEP version, and the EPOC-2 scheme. Because of the new 512-bits record for factorization, we consider 1024-bit moduli (even for EPOC-2, where the modulus is not of the classical format, and may be easier to factor). Since related messages attacks [10] cannot be proceeded on all those RSA-based scheme, excepted the original one, we will use $e = 3$. For OAEP, we take $k_1 = 64$. For OTP–Dependent RSA, we take $k_2 = 160$.

Schemes $\lvert N \rvert = 1024$	**RSA**	**OAEP**	**EPOC-2**	**OTP-DRSA**
Security				
One-Wayness	RSA	RSA	Fact	RSA
Semantic Security	–	RSA	Fact	RSA
Chosen-Ciphertext Security	–	RSA	Fact	RSA
Size (in bits)				
Plaintext	1024	448	512	1024
Ciphertext	1024	1024	2048	2208
Expansion	1	2.3	4	2.2
Encryption				
Workload/Bloc	2	2	1024	4
Workload/kB	16	37	8192	32
Decryption				
Workload/Bloc	384	384	569	386
Workload/kB	3072	7022	9102	3088

Fig. 3: Efficiency of Encryptions and Decryptions

Remark 2. In this table, the basic operation is the modular multiplication with a 1024-bit long modulus. We assume that the modular multiplication algorithm is quadratic in the modulus size and that modular squares are computed with the same algorithm. Furthermore, in the decryption phase, we always use the CRT.

One can remark that the OTP–Dependent RSA scheme is the most efficient scheme, since it encrypts just a slightly faster than OAEP but decrypts twice as fast as OAEP. The comparison with EPOC-2 is clear, even for the decryption, OTP–Dependent RSA is 3 times faster.

However, both EPOC-2 and OTP–Dependent RSA can increase their efficiency integrating a *one-time secure* symmetric encryption scheme, which makes this general proposal (the Hybrid Dependent RSA scheme) the fastest encryption scheme based on factorization of integers.

5 Limitations

The proofs are performed in the random oracle model, like EPOC [21]. The limitations are therefore the same.

6 Intellectual Property Statement

Neither the CNRS and the ENS, nor the author has any patents or patent applications relevant to this proposal. The Dependent–RSA–based contributions have all been placed in the public domain.

References

[1] M. Abdalla, M. Bellare, and P. Rogaway. DHAES: An Encryption Scheme Based on the Diffie-Hellman Problem. IEEE P1363a Submission. September 1998. Available from `http://grouper.ieee.org/groups/1363/addendum.html`.

[2] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among Notions of Security for Public-Key Encryption Schemes. In *Crypto '98*, LNCS 1462, pages 26–45. Springer-Verlag, Berlin, 1998.

[3] M. Bellare and P. Rogaway. Random Oracles Are Practical: a Paradigm for Designing Efficient Protocols. In *Proc. of the 1st CCCS*, pages 62–73. ACM Press, New York, 1993.

[4] M. Bellare and P. Rogaway. Optimal Asymmetric Encryption – How to Encrypt with RSA. In *Eurocrypt '94*, LNCS 950, pages 92–111. Springer-Verlag, Berlin, 1995.

[5] M. Bellare and A. Sahai. Non-Malleable Encryption: Equivalence between Two Notions, and an Indistinguishability-Based Characterization. In *Crypto '99*, LNCS 1666, pages 519–536. Springer-Verlag, Berlin, 1999.

[6] D. Bleichenbacher. A Chosen Ciphertext Attack against Protocols based on the RSA Encryption Standard PKCS #1. In *Crypto '98*, LNCS 1462, pages 1–12. Springer-Verlag, Berlin, 1998.

[7] V. Boyko, M. Peinado, and R. Venkatesan. Speedings up Discrete Log and Factoring Based Schemes via Precomputations. In *Eurocrypt '98*, LNCS 1403. Springer-Verlag, Berlin, 1998.

[8] S. A. Brands. An Efficient Off-Line Electronic Cash System Based on the Representation Problem. Technical Report CS-R9323, CWI, Amsterdam, 1993.

[9] R. Canetti, O. Goldreich, and S. Halevi. The Random Oracles Methodology, Revisited. In *Proc. of the 30th STOC*, pages 209–218. ACM Press, New York, 1998.

[10] D. Coppersmith, M. Franklin, J. Patarin, and M. Reiter. Low-Exponent RSA with Related Messages. In *Eurocrypt '96*, LNCS 1070, pages 1–9. Springer-Verlag, Berlin, 1996.

[11] D. Coppersmith, S. Halevi, and C. S. Jutla. ISO 9796 and the New Forgery Strategy. Working Draft presented at the Rump Session of Crypto '99, 1999.

[12] S. Coron, D. Naccache, and Ju. Stern. On the Security of RSA Padding. In *Crypto '99*, LNCS 1666, pages 1–18. Springer-Verlag, Berlin, 1999.

[13] R. Cramer and V. Shoup. A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack. In *Crypto '98*, LNCS 1462, pages 13–25. Springer-Verlag, Berlin, 1998.

[14] W. Diffie and M. E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT–22(6):644–654, November 1976.

[15] D. Dolev, C. Dwork, and M. Naor. Non-Malleable Cryptography. In *Proc. of the 23rd STOC*. ACM Press, New York, 1991.

[16] E. Fujisaki and T. Okamoto. How to Enhance the Security of Public-Key Encryption at Minimum Cost. In *PKC '99*, LNCS 1560, pages 53–68. Springer-Verlag, Berlin, 1999.

[17] E. Fujisaki and T. Okamoto. Secure Integration of Asymmetric and Symmetric Encryption Schemes. In *Crypto '99*, LNCS 1666, pages 537–554. Springer-Verlag, Berlin, 1999.

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

[19] J. Håstad. Solving Simultaneous Modular Equations of Low Degree. *SIAM Journal of Computing*, 17:336–341, 1988.

[20] M. Naor and M. Yung. Public-Key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. In *Proc. of the 22nd STOC*, pages 427–437. ACM Press, New York, 1990.

[21] T. Okamoto, S. Uchiyama, and E. Fujisaki. EPOC: Efficient Probabilistic Public-Key Encryption. Submission to IEEE P1363a. November 1998.
Available from `http://grouper.ieee.org/groups/1363/addendum.html`.

[22] T. Okamoto, E. Fujisaki and H. Morita. PSEC: Provably Secure Elliptic Curve Encryption Scheme. Submission to IEEE P1363a. March 1999. Available from `http://grouper.ieee.org/groups/1363/addendum.html`.

[23] D. Pointcheval. New Public Key Cryptosystems based on the Dependent-RSA Problems. In *Eurocrypt '99*, LNCS 1592, pages 239–254. Springer-Verlag, Berlin, 1999. (This paper is proposed in the appendix).

[24] D. Pointcheval. Chosen-Ciphertext Security for any One-Way Cryptosystem. In *PKC '00*, LNCS. Springer-Verlag. Berlin, 2000.

[25] C. Rackoff and D. R. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In *Crypto '91*, LNCS 576, pages 433–444. Springer-Verlag, Berlin, 1992.

[26] R. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.

[27] RSA Data Security, Inc. Public Key Cryptography Standards – PKCS. Available from `http://www.rsa.com/rsalabs/pubs/PKCS/`.

[28] V. Strassen. The Computational Complexity of Continued Fractions. *SIAM Journal of Computing*, 12(1):1–27, 1983.

[29] Y. Tsiounis and M. Yung. On the Security of El Gamal based Encryption. In *PKC '98*, LNCS. Springer-Verlag, Berlin, 1998.

Appendix: New Public Key Cryptosystems based on the Dependent–RSA Problems

This paper has been presented at Eurocrypt '99 – Prague, Czech Republic, and published by Springer-Verlag, Berlin, in the Lecture Notes in Computer Science, Volume 1592, pages 239–254.

Advances in Cryptology – Proceedings of EUROCRYPT '99 (2 – 6 may 1999, Prague, Czech Republic)
J. Stern Ed. Springer-Verlag, LNCS 1592, pages 239–254.

New Public Key Cryptosystems based on the Dependent–RSA Problems

David Pointcheval

LIENS – CNRS, École Normale Supérieure,
45 rue d'Ulm, 75230 Paris Cedex 05, France.

E-mail: David.Pointcheval@ens.fr,
URL: http://www.dmi.ens.fr/~pointche.

May 1999

Abstract.

Since the Diffie-Hellman paper, asymmetric encryption has been a very important topic, and furthermore ever well studied. However, between the efficiency of RSA and the security of some less efficient schemes, no trade-off has ever been provided.
In this paper, we propose better than a trade-off: indeed, we first present a new problem, derived from the RSA assumption, the "Dependent–RSA Problem". A careful study of its difficulty is performed and some variants are proposed, namely the "Decisional Dependent–RSA Problem".
They are next used to provide new encryption schemes which are both secure and efficient. More precisely, the main scheme is proven semantically secure in the standard model. Then, two variants are derived with improved security properties, namely against adaptive chosen-ciphertext attacks, in the random oracle model. Furthermore, all those schemes are more or less as efficient as the original RSA encryption scheme and reach semantic security.

Keywords: Public-Key Encryption, Semantic Security, Chosen-Ciphertext Attacks, the Dependent–RSA Problem

Introduction

Since the seminal Diffie-Hellman paper [9], which presented the foundations of the asymmetric cryptography, public-key cryptosystems have been an important goal for many people. In 1978, the RSA cryptosystem [20] was the first application and remains the most popular scheme. However, it does not satisfy any security criterion (*e.g.*, the RSA encryption standard PKCS #1 v1.5 has even been recently broken [4]) and was subject to numerous attacks (broadcast [13], related messages [7], etc).

Notions of Security. In 1984, Goldwasser and Micali [12] defined some security notions that an encryption scheme should satisfy, namely *indistinguishability of encryptions* (a.k.a. *polynomial security* or *semantic security*). This notion means that a ciphertext does not leak any useful information about the plaintext, but its length, to a polynomial time attacker. For example, if an attacker knows that the plaintext is either "sell" or "buy", the ciphertext does not help him.

By the meantime, El Gamal [11] proposed a probabilistic encryption scheme based on the Diffie-Hellman problem [9]. Its semantic security, relative to the Decisional Diffie-Hellman problem, was formally proven just last year [23], even if the result was informally well known. However this scheme never got very popular because of its computational load.

During the last ten years, beyond semantic security, a new security notion has been defined: the *non-malleability* [10]. Moreover, some stronger scenarios of attacks have been considered: the *(adaptive) chosen-ciphertext attacks* [16, 19]. More precisely, the non-malleability property means that any attacker cannot modify a ciphertext while keeping any control over the relation between the resulting plaintext and the original one. On the other hand, the stronger scenarios give partial or total access to a decryption oracle to the attacker (against the semantic security or the non-malleability). Another kind of property for encryption schemes has also been defined, called *Plaintext-Awareness* [3], which means that no one can produce a valid ciphertext without knowing the corresponding plaintext. At last Crypto, Bellare *et al.* [1] provided a precise analysis of all these security notions. The main practical result is the equivalence between non-malleability and semantic security in adaptive chosen-ciphertext scenarios.

New Encryption Schemes. Besides all these strong notions of security, very few new schemes have been proposed. In 1994, Bellare and Rogaway [3] presented some variants of RSA semantically secure even in the strong sense (*i.e.* against adaptive chosen-ciphertext attacks) in the random oracle model [2]. But we had to wait 1998 to see other practical schemes with proofs of semantic security: Okamoto–Uchiyama [17], Naccache–Stern [15] and Paillier [18] all based on higher residues; Cramer–Shoup [8] based on the Decisional Diffie-Hellman problem. Nevertheless, they remain rather inefficient. Indeed, all of them are in a discrete logarithm setting and require many full-size exponentiations for the encryption process. Therefore, they are not more efficient than the El Gamal encryption scheme.

The random oracle model. The best security argument for a cryptographic protocol is a proof in the standard model relative to a well-studied difficult problem, such as RSA, the factorization or the discrete logarithm. But no really efficient cryptosystem can aspire to such an argument. Indeed, the best encryption scheme that achieves chosen-ciphertext security in the standard model was published last year [8], and still requires more than four exponentiations for an encryption.

In 1993, Bellare and Rogaway [2] defined a model, the so-called "Random Oracle Model", where some objects are idealized, namely hash functions which are assumed perfectly random. This helped them to design later OAEP [3], the most efficient encryption scheme known until now. In spite of a recent paper [6] making people to be careful with the random oracle model, the security of OAEP has been widely agreed. Indeed, this scheme is incorporated in SET, the Secure Electronic Transaction system [14] proposed by VISA and MasterCard, and will become the new RSA encryption standard PKCS #1 v2.0 [21].

Furthermore, an important feature of the random oracle model is to provide efficient reductions between a well-studied mathematical problem and an attack. Therefore, the reduction validates protocols together with practical parameters. Whereas huge-polynomial reductions, which can hardly be avoided in the standard model, only prove asymptotic security, for large parameters.

As a conclusion, it is better to get an efficient reduction in the random oracle model than a complex reduction in the standard model, since this latter does not prove anything for practical sizes!

Aim of our work. Because of all these inefficient or insecure schemes, it is clear that, from now, the main goal is to design a cryptosystem that combines both efficiency and security. In other words, we would like a *semantically secure scheme as efficient as RSA*.

Outline of the paper. Our feeling was that such a goal required new algebraic problems. In this paper, we first present the *Computational Dependent–RSA problem*, a problem derived from the RSA assumption. We also propose a decisional variant, the *Decisional Dependent–RSA problem*. Then, we give some arguments to validate the cryptographic purpose of those problems, with a careful study of their difficulty and their relations with RSA. Namely, the Computational Dependent–RSA problem is, in a way, equivalent to RSA.

Next, we apply them successfully to the asymmetric encryption setting, and we present a very efficient encryption scheme with the proof of its *semantic security* relative to the *Decisional Dependent–RSA problem* in the standard model. Thereafter, we present two techniques to make this scheme semantically secure both *against adaptive chosen-ciphertext attacks* and relative to the *Computational Dependent–RSA problem* in the random oracle model. Both techniques improve the security level at a very low cost.

1 The Dependent–RSA Problems

As claimed above, the only way to provide new interesting encryption schemes seems to find new algebraic problems. In this section, we focus on new problems with a careful study of both their difficulty and their relations.

1.1 Definitions

For all the problems presented below, we are given a large composite RSA modulus N and an exponent e relatively prime to $\varphi(N)$, the totient function of the modulus N. Let us define a first new problem called the *Computational Dependent–RSA Problem* (C–DRSA).

Definition 4 (The Computational Dependent–RSA: C–DRSA(N, e)).

Given: $\alpha \in \mathbb{Z}_N^\star$;

Find: $(a+1)^e \bmod N$, where $\alpha = a^e \bmod N$.

Notation: We denote by $\mathsf{Succ}(\mathcal{A})$ the success probability of an adversary $\mathcal{A}$:

$$\mathsf{Succ}(\mathcal{A}) = \Pr\left[\mathcal{A}(a^e \bmod N) = (a+1)^e \bmod N \,\middle|\, a \stackrel{R}{\leftarrow} \mathbb{Z}_N^\star\right].$$

As it has already been done with the Diffie-Hellman problem [9], we can define a decisional version of this problem, therefore called the *Decisional Dependent–RSA Problem* (D–DRSA): Given a candidate to the Computational Dependent–RSA problem, is it the right solution? This decisional variant will then lead to a semantically secure encryption scheme.

Definition 5 (The Decisional Dependent–RSA: D–DRSA(N, e)).

Problem: Distinguish the two distributions

$$\mathcal{R}and = \left\{(\alpha,\gamma) = (a^e \bmod N, c^e \bmod N) \,\middle|\, a, c \stackrel{R}{\leftarrow} \mathbb{Z}_N^\star\right\},$$
$$\mathcal{DRSA} = \left\{(\alpha,\gamma) = (a^e \bmod N, (a+1)^e \bmod N) \,\middle|\, a \stackrel{R}{\leftarrow} \mathbb{Z}_N^\star\right\}.$$

Notation: We denote by $\mathsf{Adv}(\mathcal{A})$ the advantage of a distinguisher $\mathcal{A}$:

$$\mathsf{Adv}(\mathcal{A}) = \left| \Pr_{\mathcal{R}and}[\mathcal{A}(\alpha,\gamma)=1] - \Pr_{\mathcal{DRSA}}[\mathcal{A}(\alpha,\gamma)=1] \right|.$$

1.2 The Dependent–RSA Problems and RSA

In order to study those Dependent–RSA problems, we define a new one, we call the *Extraction Dependent–RSA Problem* (E–DRSA):

Given: $\alpha = a^e \in \mathbb{Z}_N^\star$ and $\gamma = (a+1)^e \in \mathbb{Z}_N^\star$;

Find: $a \bmod N$.

One can then prove that extraction of e-th roots is easier to solve than the Computational Dependent–RSA problem and the Extraction Dependent–RSA problem together.

Theorem 1. $\mathbf{RSA(N, e)} \Longleftrightarrow$ *E–DRSA*$(\mathbf{N}, \mathbf{e})$ + *C–DRSA*$(\mathbf{N}, \mathbf{e})$.

Proof. Let $\mathcal{A}$ be an E–DRSA adversary and $\mathcal{B}$ a C–DRSA adversary. For a given $c = a^e \bmod N$, an element of $\mathbb{Z}_N^\star$, whose e-th root is wanted, one uses $\mathcal{B}$ to obtain $(a+1)^e \bmod N$ and gets a from $\mathcal{A}(a^e \bmod N, (a+1)^e \bmod N)$.

The opposite direction is trivial, since extraction of e-th roots helps to solve all the given problems. □

Furthermore, it is clear that any decisional problem is easier to solve than its related computational version, and trying to extract a, it is easy to decide whether the given γ is the right one. Finally, for any (N, e), the global picture is

$$\text{C–DRSA} + \text{E–DRSA} \Longleftrightarrow \mathbf{RSA} \Longrightarrow \text{C–DRSA}, \text{E–DRSA} \Longrightarrow \text{D–DRSA},$$

where $A \Longrightarrow B$ means that an oracle that breaks A can be used to break B within a time polynomial in the size of N.

2 How to Solve the Dependent–RSA Problems?

In order to use these problems in cryptography, we need to know their practical difficulty, for reasonable sizes. Hopefully, some of them have already been studied in the past. Indeed, they are related to many properties of the RSA cryptosystem, namely its malleability, its security against related-message attacks [7] and in the multicast setting [13].

Concerning the Extraction Dependent–RSA problem, some methods have been proposed by Coppersmith *et al.* [7], trying to solve the related-message system:

$$\begin{cases} \alpha = m^e \bmod N \\ \beta = (m+1)^e \bmod N \end{cases}$$

2.1 A First Method: Successive Eliminations

Let us assume that $e = 3$, then it is possible to successively eliminate the powers of m and express m from α and β:

$$\begin{cases} \alpha = m^3 \bmod N \\ \beta = (m+1)^3 = m^3 + 3m^2 + 3m + 1 \bmod N \\ \quad = \alpha + 3m^2 + 3m + 1 \bmod N \end{cases}$$

$$\begin{cases} m \times (\beta - \alpha) - 3\alpha = 3m^2 + m \bmod N \\ \qquad\qquad \beta - \alpha = (3m^2 + m) + 2m + 1 \bmod N \\ \qquad\qquad\qquad = m \times (\beta - \alpha + 2) - 3\alpha + 1 \bmod N \end{cases}$$

$$\text{Then, } m = \frac{2\alpha + \beta - 1}{\beta - \alpha + 2} \bmod N.$$

First, Coppersmith *et al.* [7] claimed that for each e, there exist polynomials P and Q such that each can be expressed as rational polynomials in X^e and $(X+1)^e$, and such that $Q(X) = XP(X)$. Then $m = Q(m)/P(m)$. However, the explicit expression of m as a ratio of two polynomials in α and β requires $\Theta(e^2)$ coefficients, furthermore it is not obvious how to calculate them efficiently.

Consequently, this first method fails as soon as e is greater than, say 2^{40}.

2.2 A Second Method: Greatest Common Divisor

A second method comes from the remark that m is a root for both the polynomials P and Q over the ring $\mathbb{Z}_N$, where.

$$P(X) = X^e - \alpha \text{ and } Q(X) = (X+1)^e - \beta.$$

Then $X - m$ is a divisor of the gcd of P and Q. Furthermore, one can see that with high probability, it is exactly the gcd. A straightforward implementation of Euclid's algorithm takes $\mathcal{O}(e^2)$ operations in the ring $\mathbb{Z}_N$. More sophisticated techniques can be used to compute the gcd in $\mathcal{O}(e \log^2 e)$ time [22]. Then, this second method fails as soon as e is greater than 2^{60}.

2.3 Consequences on the Computational Dependent–RSA problem

Since the RSA cryptosystem appeared [20], many people have attempted to find weaknesses. Concerning the malleability of the encryption, the multiplicative property is well-known. In other words, it is easy to derive the encryption of $m \times m'$ from the encryption of m, for any m', without knowing the message m itself. However, from the encryption of an unknown message m, nothing has been found to derive the encryption of $m + 1$ whatever the exponent e may be.

Concerning the Extraction Dependent–RSA problem, one can then state the following theorem:

Theorem 1. *There exist algorithms that solve the problem* $\mathsf{E\text{–}DRSA}(N, e)$ *in* $\mathcal{O}(|N|^2, e \times |e|^2)$ *time.*

In conjunction with the Theorem 1, we can therefore claim that

Theorem 2. *There exists a reduction from the RSA problem to the Computational Dependent–RSA problem in* $\mathcal{O}(|N|^2, e \times |e|^2)$ *time.*

Then, for any fixed exponent e, $RSA(N, e)$ is reducible to $\mathsf{C\text{–}DRSA}(N, e)$ polynomially in the size of N, since the Extraction Dependent–RSA problem is "easy" to solve, using the gcd technique (see the previous version).

Anyway, computation of e-th roots seems always required to solve the Computational Dependent–RSA problem, which is intractable for any exponent e, according to the RSA assumption.

Conjecture 3. The Computational Dependent–RSA problem is intractable for large enough RSA moduli.

Remark 3. Because of the Theorem 2, this conjecture holds for small exponents, since then $\mathsf{C\text{–}DRSA}$ is as hard as RSA.

2.4 About the Decisional Dependent–RSA intractability

The gcd technique seems to be the best known attack against the Decisional Dependent–RSA problem and is impractical as soon as the exponent e is greater than 2^{60}. Which leads to the following conjecture:

Conjecture 4. The Decisional Dependent–RSA problem is intractable as soon as the exponent e is greater than 2^{60}, for large enough RSA moduli.

3 Security Notions for Encryption Schemes

For the formal definitions of all the kinds of attacks and of security notions, we refer the reader to the last Crypto paper [1]. However, let us briefly recall the main security notion, the *semantic security* (a.k.a. *indistinguishability of encryptions*) defined by Goldwasser and Micali [12]. For this notion, an attacker is seen as a two-stage ("find-and-guess") Turing machine which first chooses two messages, during the "find"-stage. In the second

Initialization
$N = pq$, a large RSA modulus
e, an exponent, relatively prime to $\varphi(N)$
Public key: (N, e)
Secret key: $d = e^{-1} \bmod \varphi(N)$

Encryption of $m \in \{0, \ldots, N-1\}$
$k \in_R \mathbb{Z}_N^\star$
$A = k^e \bmod N$
$B = m \times (k+1)^e \bmod N$
Then, $C = (A, B)$.

Decryption of $C = (A, B)$
$k = A^d \bmod N$
$m = B/(k+1)^e \bmod N$

Fig. 1: The DRSA Encryption Scheme

stage, the "guess"-stage, she receives a challenge, which is the encryption of one of both chosen messages, and has to guess which one is the corresponding plaintext.

In the public-key setting, any attacker can play a *chosen-plaintext attack*, since she can encrypt any message she wants. However, stronger attacks has been defined. First, Naor and Yung [16] defined the *chosen-ciphertext attack* (a.k.a. *lunchtime attack*) where the attacker has access to a decryption oracle during the "find"-stage, to choose the two plaintexts. Then, Rackoff and Simon [19] improved this notion, giving the decryption oracle access to the attacker in both stages (with the trivial restriction not to ask the challenge ciphertext). This attack is known as *adaptive chosen-ciphertext attack* and is the strongest that an attacker can play, in the classical model.

The aim of this paper is to provide a new efficient scheme, semantically secure against adaptive chosen-ciphertext attacks.

4 The DRSA Encryption Scheme

The Dependent–RSA problem can be used, like the Diffie-Hellman problem [9], to provide encryption schemes. An RSA version of the El Gamal encryption [11] is then proposed with some security properties, namely semantic security against chosen-plaintext attacks. In the next section, we propose two variants with very interesting improved security properties together with high efficiency.

4.1 Description

The scheme works as described in figure 1. We are in the RSA setting: each user publishes an RSA modulus N while keeping secret the prime factors p and q. He also chooses a public exponent e and its inverse d modulo $\varphi(N)$. The public key consists in the pair (N, e), while the secret key is the private exponent d (it can also consists in the prime factors p and q to improve the decryption algorithm efficiency, using the Chinese Remainders Theorem). To encrypt the message $m \in \{0, \dots, N-1\}$ to Alice whose public key is (N, e), Bob chooses a random $k \in \mathbb{Z}_N^\star$ and computes $A = k^e \bmod N$ as well as $B = m \times (k+1)^e \bmod N$. He sends the pair (A, B) to Alice. When she receives a pair (A, B), Alice computes $k = A^d \bmod N$ and recovers the plaintext $m = B/(k+1)^e \bmod N$.

4.2 Security Properties

The same way as for the El Gamal encryption scheme, one can prove the semantic security of this scheme.

Theorem 1. *The DRSA encryption scheme is semantically secure against chosen-plaintext attacks relative to the Decisional Dependent–RSA problem.*

Proof. Let us consider an attacker $\mathcal{A} = (A_1, A_2)$ who can break the semantic security of this scheme within a time t and with an advantage, in the "guess"-stage, greater than ε.

In the figure beside, we construct a D–DRSA adversary, $\mathcal{B}$, who is able to break the Decisional Dependent–RSA problem for the given public key (N, e) with an advantage greater than $\varepsilon/2$ and a similar running time. The equivalence between the semantic security and the Decisional Dependent–RSA problem will follow, since the opposite direction is straightforward.

```
B(α, γ):
    Run A_1(pk)
        Get m_0, m_1, s
    Randomly choose b ∈ {0,1}
    A = α, B = m_b · γ mod N
    Run A_2(s, m_0, m_1, (A, B))
        Get c
    if c = b Return 1
    else Return 0
```

On one hand, we have to study the probability for A_2 to answer $c = b$ when the pair (α, γ) comes from the random distribution. But in this case, one can see that the pair $(A, B) \in \{(r^e, m_b s^e) \mid r, s \in \mathbb{Z}_N^\star\}$ is uniformly distributed in the product space $\mathbb{Z}_N^\star \times \mathbb{Z}_N^\star$, hence independently of b. Then

$$\Pr_{\mathcal{R}and}[\mathcal{B}(\alpha, \gamma) = 1] = \Pr_{\mathcal{R}and}[c = b] = \frac{1}{2}.$$

On the other hand, when the pair (α, γ) comes from the $\mathcal{DRSA}$ distribution, one can remark that (A, B) is a valid ciphertext of m_b, following a uniform distribution among the possible ciphertexts. Then

$$\Pr_{\mathcal{DRSA}}[\mathcal{B}(\alpha, \gamma) = 1] = \Pr_{\mathcal{DRSA}}[c = b] = \Pr_b[A_2(s, m_0, m_1, \mathcal{E}(m_b)) = b] \stackrel{def}{=} \frac{1}{2} \pm \frac{\mathsf{Adv}^{\mathcal{A}}}{2}.$$

The advantage of $\mathcal{B}$ in distinguishing the $\mathcal{DRSA}$ and the $\mathcal{R}and$ distributions is $\mathsf{Adv}(\mathcal{B}) = \mathsf{Adv}^{\mathcal{A}}/2$, and therefore greater than $\varepsilon/2$. □

5 Some Variants

As it has already been remarked, attackers can be in a stronger scenario than the chosen-plaintext one. Now, we improve the security level, making the scheme resistant to adaptive

Initialization
ℓ, security parameter
$N = pq$, a large RSA modulus
e, an exponent, relatively prime to $\varphi(N)$
$h : \mathbb{Z}_N \times \mathbb{Z}_N \to \{0,1\}^\ell$, a hash function
Public key: (N, e)
Secret key: $d = e^{-1} \bmod \varphi(N)$

Encryption of $m \in \{0, \ldots, N-1\}$
$k \in_R \mathbb{Z}_N^\star$
$A = k^e \bmod N$
$B = m \times (k+1)^e \bmod N$
$H = h(m, k) \in \{0,1\}^\ell$
Then, $C = (A, B, H)$

Decryption of $C = (A, B, H)$
$k = A^d \bmod N$
$m = B/(k+1)^e \bmod N$
$H \stackrel{?}{=} h(m, k)$

Fig. 2: First Variant: The DRSA-1 Encryption Scheme

chosen-ciphertext attacks, in the random oracle model. In a second step, we weaken the algorithmic assumption: an attacker against the semantic security of the second variant, in an adaptive chosen-ciphertext scenario, can be used to efficiently break the Computational Dependent–RSA problem, and not only the Decisional Dependent–RSA problem.

Furthermore, it is important to remark that both improvements are very low-cost on both a computational point of view and the size of the ciphertexts.

5.1 Description of the First Variant: DRSA-1

The scheme works as described in figure 2, where h is a hash function, seen like a random oracle which outputs ℓ-bit numbers. The initialization is unchanged. To encrypt a message $m \in \{0, \ldots, N-1\}$ to Alice whose public key is (N, e), Bob chooses a random $k \in \mathbb{Z}_N^\star$ and computes $A = k^e \bmod N$ as well as $B = m \times (k+1)^e \bmod N$ and the control padding $H = h(m, k)$. He sends the triple (A, B, H) to Alice. When she receives a triple (A, B, H), Alice first computes the random value $k = A^d \bmod N$ and recovers the probable plaintext $m = B/(k+1)^e \bmod N$. She then checks whether they both satisfy the control padding $H = h(m, k)$.

5.2 Security Properties

Concerning this scheme, we claim the following result:

Theorem 1. *The DRSA-1 encryption scheme is semantically secure against adaptive chosen-ciphertext attacks relative to the Decisional Dependent–RSA problem in the random oracle model.*

Proof. This proof is similar to the previous one except two simulations. Indeed, we first have to simulate the random oracle, and more particularly for the challenge ciphertext, which is the triple $(A = \alpha, B = m_b \times \gamma, H)$, where H is randomly chosen in $\{0,1\}^\ell$. But for any new query to the random oracle, one simply returns a new random value. Furthermore, any query (m, k) to the random oracle is filtered: if $k^e = \alpha \bmod N$, then we stop the game, and whether $\gamma = (k+1)^e \bmod N$ we output 1 or 0. Secondly, since we are in an adaptive chosen-ciphertext scenario, we have to simulate the decryption oracle: when the adversary asks a query (A', B', H'), the simulator looks in the table of the queries previously made to the random oracle to find the answer H'. Then, two cases may appear:

- H' has been returned by the random oracle and corresponds to a query (m, k) (there may be many queries corresponding to this answer). The simulator checks whether $A' = k^e \bmod N$ and $B' = m \times (k+1)^e \bmod N$. Then it returns m as the decryption of the triple (A', B', H'). Otherwise, the simulator considers that it is an invalid ciphertext and returns the reject symbol "*".
- Otherwise, the simulator returns the reject symbol "*".

The bias is the same as above when all the simulations are correctly made. Concerning the simulation of the random oracle, it is perfectly made, because of the randomness of the answers. However, some decryptions may be incorrect, but only refusing a valid ciphertext: a ciphertext is refused if the query (m, k) has not been asked to the random oracle h. However, the attacker might have guessed the right value for $h(m, k)$ without having asked for it, but only with probability $1/2^\ell$.

Then, if the pair (α, γ) comes from the $\mathcal{DRSA}$ distribution, since the probability of success can be improved if the adversary guesses the e-th root of α, which had led to stop the game with an answer 1,

$$\Pr_{\mathcal{DRSA}}[\mathcal{B}(\alpha,\gamma) = 1] \geq \frac{1}{2} + \frac{\mathsf{Adv}^{\mathcal{A}}}{2} - \frac{q_d}{2^\ell},$$

where the adversary asks at most q_d queries to the decryption oracle. However, if the pair (α, γ) comes from the random distribution, for the same reason as in the previous proof, the adversary cannot gain any advantage, except the case where she had guessed the e-th root of α, but then, $\mathcal{B}$ likely outputs 0:

$$\Pr_{\mathcal{R}and}[\mathcal{B}(\alpha,\gamma) = 1] \leq \frac{1}{2} - \Pr[\alpha^d \text{ guessed}] \leq \frac{1}{2}.$$

Therefore, $\mathsf{Adv}(\mathcal{B}) \geq \dfrac{\mathsf{Adv}^{\mathcal{A}}}{2} - \dfrac{q_d}{2^\ell}$. □

5.3 Description of the Second Variant: DRSA-2

We can furthermore weaken the algorithmic assumption, making the scheme equivalent to the computational problem rather than to the decisional one. The variant works as

Initialization
k_1, size of the plaintext
k_2, security parameter
$N = pq$, a large RSA modulus
e, an exponent, relatively prime to $\varphi(N)$
$h_1 : \mathbb{Z}_N \to \{0,1\}^{k_1}$, a hash function
$h_2 : \{0,1\}^{k_1} \times \mathbb{Z}_N \to \{0,1\}^{k_2}$, a hash function
Public key: (N, e)
Secret key: $d = e^{-1} \bmod \varphi(N)$

Encryption of $m \in \{0,1\}^{k_1}$
$k \in_R \mathbb{Z}_N^\star$
$A = k^e \bmod N$
$B = m \oplus h_1((k+1)^e \bmod N)$
$H = h_2(m, k)$
Then, $C = (A, B, H)$

Decryption of $C = (A, B, H)$
$k = A^d \bmod N$
$m = B \oplus h_1((k+1)^e \bmod N)$
$H \stackrel{?}{=} h_2(m, k)$

Fig. 3: Second Variant: The DRSA-2 Encryption Scheme

described in figure 3, where h_1 and h_2 are two hash functions, seen like random oracles which output k_1-bit numbers and k_2-bit numbers respectively. The initialization is unchanged. To encrypt a message $m \in \{0,1\}^{k_1}$ to Alice whose public key is (N, e), Bob chooses a random $k \in \mathbb{Z}_N^\star$ and computes $A = k^e \bmod N$. He can then mask the message in $B = m \oplus h_1((k+1)^e \bmod N)$, a k_1-bit long string and compute the control padding $H = h_2(m, k) \in \{0,1\}^{k_2}$. He sends the triple (A, B, H) to Alice. When she receives a ciphertext (A, B, H), Alice first computes the random value $k = A^d \bmod N$. She can therefore recover the probable plaintext $m = B \oplus h_1((k+1)^e \bmod N)$. Then, she checks whether they both satisfy the control padding, $H = h_2(m, k)$.

Theorem 2. *The DRSA-2 encryption scheme is semantically secure against adaptive chosen-ciphertext attacks relative to the Dependent–RSA problem in the random oracle model.*

Proof. The result comes from the fact that any attacker cannot gain any advantage in distinguishing the original plaintext (in an information theoretical sense) if she has not asked for any $(\star, k)$ to h_2 (which is called "event 1" and denoted by E_1) or for $(k+1)^e \bmod N$ to h_1 (which is called "event 2" and denoted by E_2). Then, for a given $\alpha = a^e \bmod N$, either we learn the e-th root of α, or $(a+1)^e \bmod N$ is in the list of the queries asked to h_1. Both cases lead to the computation of $(a+1)^e \bmod N$.

More precisely, let $\mathcal{A} = (A_1, A_2)$ be an attacker against the semantic security of the DRSA-2 encryption scheme, using an adaptive chosen-ciphertext attacker. Within a time bound t, she asks q_d queries to the decryption oracle and q_h queries to the random oracles and distinguishes the right plaintext with an advantage greater than ε. We can use her to provide an algorithm that solves the Computational Dependent–RSA problem, simply filtering the queries asked to the random oracles.

Actually, because of the randomness of the random oracle h_1, if no critical queries have been asked,

$$\begin{aligned} &\Pr_b[A_2(s, m_0, m_1, \mathcal{E}(m_b)) = b] = \frac{1}{2} \pm \frac{\mathsf{Adv}^{\mathcal{A}}}{2} \\ &= \Pr_b[A_2 = b \wedge \neg(\mathsf{E}_1 \vee \mathsf{E}_2)] + \Pr_b[A_2 = b \wedge (\mathsf{E}_1 \vee \mathsf{E}_2)] \\ &= \Pr[\neg(\mathsf{E}_1 \vee \mathsf{E}_2)] \times 1/2 + \Pr_b[A_2 = b \wedge (\mathsf{E}_1 \vee \mathsf{E}_2)]. \end{aligned}$$

Then, $\pm\mathsf{Adv}^{\mathcal{A}} = \Pr[\mathsf{E}_1 \vee \mathsf{E}_2] - 2 \times \Pr_b[A_2(s, m_0, m_1, \mathcal{E}(m_b)) = b \wedge (\mathsf{E}_1 \vee \mathsf{E}_2)]$, and both cases imply $\Pr[\mathsf{E}_1 \vee \mathsf{E}_2] \geq \mathsf{Adv}^{\mathcal{A}}$.

Using our simulations, namely for the decryption oracle, we obtain, as previously seen,

$$\Pr[(\mathsf{E}_1 \vee \mathsf{E}_2) \wedge \text{ no incorrect decryption}] \geq \mathsf{Adv}^{\mathcal{A}} - q_d \times 2^{-k_2}.$$

For the reduction, one just has to randomly choose the query which should correspond to $(a+1)^e \bmod N$. With probability greater than $1/q_h$, it is a good choice (or maybe, event 2 happens, but we assume the worst case). Then, with probability greater than $(\mathsf{Adv}^{\mathcal{A}} - q_d/2^{k_2})/q_h$, within roughly the same running time as the adversary $\mathcal{A}$, one obtains the right value for $(a+1)^e \bmod N$ corresponding to the given $\alpha = a^e \bmod N$. □

6 Efficiency

Now that we know that these schemes are provably secure, let us compare them with other well-known cryptosystems from a computational point of view. And first, let us briefly recall the three other schemes we will consider:

El Gamal. An authority chooses and publishes two large prime numbers p and q such that q is a large prime factor of $p-1$, together with an element g of $\mathbb{Z}_p^\star$ of order q. Each user chooses a secret key x in $\mathbb{Z}_q^\star$ and publishes $y = g^x \bmod p$. To encrypt a message m, one has to choose a random element k in $\mathbb{Z}_q^\star$ and sends the pair $(r = g^k \bmod p, s = m \times y^k \bmod p)$ as the ciphertext. The recipient can recover the message from a pair (r, s) since $m = s/r^x \bmod p$, where x is his secret key. To reach semantic security [23], this scheme requires m to be in the subgroup generated by g. To be practical, one can choose $p = 2q + 1$, a strong prime, which consequently increases the number of multiplications to be made for an encryption. We do not consider any variant of El Gamal, since all are much heavier to implement.

RSA. Each user chooses a large RSA modulus $N = pq$ of size n together with an exponent e. He publishes both and keeps secret the private exponent $d = e^{-1} \bmod \varphi(N)$. To encrypt a message m, one just has to send the string $c = m^e \bmod N$. To recover the plaintext, the recipient computes $c^d = m \bmod N$.

Optimal Asymmetric Encryption Padding. The RSA variant, OAEP, was the most efficient scheme, from our knowledge: An authority chooses and publishes two hash functions g and h which both output $n/2$-bit strings. Each user chooses as above a public key (N, e), where N is a n-bit long RSA modulus, and keeps secret the exponent d. To encrypt a message m, one has to choose a random element r, computes $A = (m \| 0^{k_1}) \oplus g(r)$ and $B = r \oplus h(A)$ and finally sends $C = (A \| B)^e \bmod N$. The recipient can recover the message from C first computing $A \| B = C^d \bmod N$, then $r = B \oplus h(A)$ and $M = A \oplus g(r)$. If M ends with k_1 zero bits, then m is the beginning of M.

Both encryption schemes (the original RSA and OAEP) essentially require one exponentiation to the power e per encryption. And as one can remark, they depend on the message, and then has to be done online.

Precomputations. In the same vein as a last Eurocrypt paper [5], our scheme allows precomputations. Indeed, a user can precompute many pairs for a given recipient, *i.e.*, $(a^e \bmod N, (a+1)^e \bmod N)$. Then an encryption only requires one multiplication, or even a XOR. However, to be fair, in the following, we won't consider this feature.

Efficiency Comparison. One can see, on figure 4, a brief comparison table involving our schemes together with the El Gamal encryption scheme (with a 512-bit long prime $p = 2q + 1$), the RSA cryptosystem and its OAEP version. Because of the new 140-digit record for factorization, for a similar security level between factorization-based schemes and discrete logarithm-based ones, we consider 1024-bit RSA-moduli: $n = |N| = 1024$, $e = 65537 = 2^{16} + 1$, and furthermore $k_1 = 64$ for OAEP. Concerning our DRSA encryption schemes, we also use a 1024-bit long modulus N. However, whereas we can use $e = 65537$ (even smaller, such as $e = 3$, since related-message attacks seem to not be applicable) in schemes based on the Computational Dependent–RSA problem (such as the DRSA-2 scheme), we need to use a larger exponent with the Decisional Dependent–RSA-based schemes, to avoid attacks presented above against the semantic security. Then, we use $e = 2^{67} + 3$, which is a prime integer, in the DRSA and in the DRSA-1 schemes.

Remark 4. In this table, the basic operation is the modular multiplication with a 1024-bit long modulus. We assume that the modular multiplication algorithm is quadratic in the modulus size and that modular squares are computed with the same algorithm. Furthermore, in the decryption phase, we use the CRT when it is possible.

CPA-IND and CCA2-IND both follow the notations of the Bellare *et al.* paper [1] and mean the indistinguishability of encryptions (a.k.a. *semantic security*) against chosen-plaintext attacks and adaptive chosen-ciphertext attacks respectively.

One can remark that our new scheme, in its basic version (DRSA–1024 bits), can encrypt **6 times faster** than El Gamal–512 bits and decrypt in essentially the same time. Therefore, the DRSA encryption schemes becomes the most efficient scheme provably semantically secure against chosen-plaintext attacks in the standard model.

If we consider the security in the random oracle model, the DRSA-1 scheme reaches the security against adaptive chosen-ciphertext attacks with an unchanged efficiency.

However, the most interesting scheme is the DRSA-2 cryptosystem that reaches semantic security both against adaptive chosen-ciphertext attacks and relative to the Computational Dependent–RSA problem, in a situation where it is practically equivalent to

Schemes	RSA 1024	OAEP 1024	El Gamal 512	DRSA 1024	DRSA-1 1024	DRSA-2 1024	
Security							
Inversion	RSA	RSA	DH	C–DRSA	C–DRSA	C–DRSA	
CPA-IND	–	RSA*	D-DH	D–DRSA	D–DRSA*	C–DRSA*	
CCA2-IND	–	RSA*	–	–	D–DRSA*	**C–DRSA***	
Size (in bits)							
Plaintext	1024	448	511	1024	1024	1024	2048
Ciphertext	1024	1024	1024	2048	2208	2208	3232
Expansion	1	2.3	2	2	2.2	2.2	1.6
Encryption							
Workload	17	17	384	139	139	35	35
Workload/kB	136	311	6144	1112	1112	280	**140**
Decryption							
Workload	384	384	192	523	523	419	419
Workload/kB	3072	7022	3072	4184	4184	3352	**1676**

* in the random oracle model

Fig. 4: Efficiency of Encryptions and Decryptions

the RSA problem. Indeed, a smaller exponent, such as $e = 65537$ (or even 3), can be used, hence an improved efficiency is obtained: with $k_1 = |N| = 1024$, this scheme is already faster than OAEP, for both encryption and decryption. Furthermore, with larger k_1 (*e.g.* $k_1 = 2048$, such as in the last column), this scheme can reach higher rates, and even get **much faster than the original RSA encryption scheme**.

Conclusion

Therefore, we have presented three new schemes with security proofs and record efficiency. Indeed, the DRSA cryptosystem is semantically secure against chosen-plaintext attacks in the standard model, relative to a new difficult problem (the inversion problem is equivalent to RSA in many cases), with an encryption rate 6 times faster than El Gamal (with similar security levels: RSA-1024 bits vs. El Gamal-512 bits).

Next, we have presented two variants semantically secure against adaptive chosen-ciphertext attacks in the random oracle model (they can even be proven plaintext-aware [3, 1]). Furthermore, the DRSA-2 scheme is more efficient than RSA, and therefore much more efficient than OAEP, with an equivalent security, since for those parameters, the Computational Dependent–RSA problem is practically equivalent to the RSA problem.

Acknowledgments

I would like to thank the anonymous Eurocrypt '99 referees for their valuable comments and suggestions, as well as Jacques Stern for fruitful discussions.

References

[1] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations Among Notions of Security for Public-Key Encryption Schemes. In *Crypto '98*, LNCS 1462, pages 26–45. Springer-Verlag, 1998.

[2] M. Bellare and P. Rogaway. Random Oracles are Practical: a Paradigm for Designing Efficient Protocols. In *Proc. of the 1st CCCS*, pages 62–73. ACM press, 1993.

[3] M. Bellare and P. Rogaway. Optimal Asymmetric Encryption – How to Encrypt with RSA. In *Eurocrypt '94*, LNCS 950, pages 92–111. Springer-Verlag, 1995.

[4] D. Bleichenbacher. A Chosen Ciphertext Attack against Protocols based on the RSA Encryption Standard PKCS #1. In *Crypto '98*, LNCS 1462, pages 1–12. Springer-Verlag, 1998.

[5] V. Boyko, M. Peinado, and R. Venkatesan. Speedings up Discrete Log and Factoring Based Schemes via Precomputations. In *Eurocrypt '98*, LNCS 1403. Springer-Verlag, 1998.

[6] R. Canetti, O. Goldreich, and S. Halevi. The Random Oracles Methodology, Revisited. In *Proc. of the 30th STOC*. ACM Press, 1998.

[7] D. Coppersmith, M. Franklin, J. Patarin, and M. Reiter. Low-Exponent RSA with Related Messages. In *Eurocrypt '96*, LNCS 1070, pages 1–9. Springer-Verlag, 1996.

[8] R. Cramer and V. Shoup. A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack. In *Crypto '98*, LNCS 1462, pages 13–25. Springer-Verlag, 1998.

[9] W. Diffie and M. E. Hellman. New Directions in Cryptography. In *IEEE Transactions on Information Theory*, volume IT–22, no. 6, pages 644–654, November 1976.

[10] D. Dolev, C. Dwork, and M. Naor. Non-Malleable Cryptography. In *Proc. of the 23rd STOC*. ACM Press, 1991.

[11] T. El Gamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In *IEEE Transactions on Information Theory*, volume IT–31, no. 4, pages 469–472, July 1985.

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

[13] J. Håstad. Solving Simultaneous Modular Equations of Low Degree. *SIAM Journal of Computing*, 17:336–341, 1988.

[14] SET Secure Electronic Transaction LLC. SET Secure Electronic Transaction Specification – Book 3: Formal Protocol Definition, may 1997. Available from `http://www.setco.org/`.

[15] D. Naccache and J. Stern. A New Cryptosystem based on Higher Residues. In *Proc. of the 5th CCCS*, pages 59–66. ACM press, 1998.

[16] M. Naor and M. Yung. Public-Key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. In *Proc. of the 22nd STOC*, pages 427–437. ACM Press, 1990.

[17] T. Okamoto and S. Uchiyama. A New Public Key Cryptosystem as Secure as Factoring. In *Eurocrypt '98*, LNCS 1403, pages 308–318. Springer-Verlag, 1998.

[18] P. Paillier. Public-Key Cryptosystems Based on Discrete Logarithms Residues. In *Eurocrypt '99*, LNCS 1592, pages 223–238. Springer-Verlag, 1999.

[19] C. Rackoff and D. R. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In *Crypto '91*, LNCS 576, pages 433–444. Springer-Verlag, 1992.

[20] R. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.

[21] RSA Data Security, Inc. Public Key Cryptography Standards – PKCS. Available from `http://www.rsa.com/rsalabs/pubs/PKCS/`.

[22] V. Strassen. The Computational Complexity of Continued Fractions. *SIAM Journal of Computing*, 12(1):1–27, 1983.

[23] Y. Tsiounis and M. Yung. On the Security of El Gamal based Encryption. In *PKC '98*, LNCS. Springer-Verlag, 1998.