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.

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, public-key cryptography has been an important goal for many people. In 1978, the RSA cryptosystem 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 and was subject to numerous attacks (broadcast, related messages, etc).

Notions of Security. In 1984, Goldwasser and Micali defined some security notions that an encryption scheme should satisfy, namely indistinguishability of encryptions (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 proposed a probabilistic encryption scheme based on the Diffie-Hellman problem. Its semantic security, relative to the Decisional Diffie-Hellman problem, was formally proven just last year, even if the result was informally well known. However, this scheme never got very popular because of its computational load.
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 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 is that such a goal requires 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 these 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 the problems presented, we need to consider the Computational Dependent–RSA and its decisional variant.

**Definition 1 (The Computational Dependent–RSA: C–DRSA).**

Given \( a \in \mathbb{Z}_N^* \), find \( a^e \mod N \) where \( a^e = \cdot \mod N \).

**Notation**

\( \text{Succ} A \) denote the success probability of an adversary \( A \):

\[
\text{Succ} A = \Pr \left[ A \text{ outputs } a^e \mod N \mid a \leftarrow \mathbb{Z}_N^* \right] \]

As it has already been done with the Diffie-Hellman problem, we can define a decisional version of this problem, therefore called the Decisional Dependent–RSA problem: 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 2 (The Decisional Dependent–RSA: $D_{DRSA}(N, e)$).

Problem Distinguish the two distributions

\[ \mathcal{Rand} = \{ \alpha, \gamma \mid a^e \equiv c^e \pmod{N}, a, c \sim \mathbb{Z}_N^* \} \]

\[ \mathcal{DRSA} = \{ \alpha, \gamma \mid a^e \equiv (a + 1)^e \pmod{N}, a \sim \mathbb{Z}_N^* \} \]

Notation $\epsilon$ denote the advantage of a distinguisher $\mathcal{A}$

\[ \text{Adv} \mathcal{A} = \Pr_{\mathcal{Rand}}[\mathcal{A} \alpha, \gamma] - \Pr_{\mathcal{DRSA}}[\mathcal{A} \alpha, \gamma] \]

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 $\alpha \sim \mathbb{Z}_N^*$ and $\gamma = (a + 1)^e \pmod{N}$;
Find $a \pmod{N}$.

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 3. RSA $(N, e) \iff E_{DRSA}(N, e) \land C_{DRSA}(N, e)$.

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

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

\[ \square \]

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

\[ \alpha = m^e \pmod{N} \quad \beta = (m + 1)^e \pmod{N} \]

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. trying to solve the related-message system.

\[ \begin{cases} \alpha \equiv m^e \pmod{N} \\ \beta \equiv m^e \pmod{N} \end{cases} \]
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 \( \alpha \) and \( \beta \):

\[
\begin{align*}
\alpha &= m^3 \mod N \\
\beta &= m^2 \mod N
\end{align*}
\]

Then,

\[
\begin{align*}
\alpha &= m^3 \mod N \\
\beta &= m^2 \mod N
\end{align*}
\]

First Coppersmith et al. 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 \( Q(X) = X P(X) \). Then \( m = Q(m)/P(m) \). However, the explicit expression of \( m \) as a ratio of two polynomials in \( \alpha \) and \( \beta \) 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

\[
P(X) = X^e - \alpha \\
Q(X) = (X + 1)^e - \beta
\]

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 \( O(e^2) \) operations in the ring \( \mathbb{Z}_N \). More sophisticated techniques can be used to compute the gcd in \( O(e \log^2 e) \) 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 ppe red ny peop e h e tte pted to nd e nesses Concerning the e i ty of the encryption the uti p ic ti e property is e no n o other ords it is e sy to deri e the encryption of \( m \times m' \) fro the encryption of \( m \) for \( ny m' \) ithout no ing the ess ge m its e f Ho e er fro the encryption of \( nun \) no n ess ge m nothing h s een found to deri e the encryption of \( m \) h te er the e ponent e y e

Concerning the E tr ction Dependent RSA pro e one c n then st te the fo o i ng theore

**Theorem 4.** There exist algorithms that solve the problem E-DRSA \( N, e \) in \( \mathcal{O} |N|^2, e \times |e|^2 \) time.
In conjunction with Theorem 3, we can therefore claim that there exists a reduction from the RSA problem to the Computational Dependent–RSA problem in $O(|N|^2, e \times |e|^2)$ time.

Then, for any fixed exponent $e$, RSA($N, e$) is reducible to $C_{DRSA}(N, e)$ 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 6. The Computational Dependent–RSA problem is intractable for large enough RSA moduli.

Remark 7. Because of the Theorem 5, this conjecture holds for small exponents, since then $C_{DRSA}$ 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 8. 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 stage, the "guess"-stage, she receives one of the 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 have 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.
Initialization

\[ N = pq, \text{ a large RSA modulus} \]
\[ e, \text{ an exponent, relatively prime to } \varphi(N) \]

**Public key:** \((N, e)\)

**Secret key:** \(d = e^{-1} \mod \varphi(N)\)

---

**Encryption of** \(m \in \mathbb{Z}_N^*\)

\[ k \in \mathbb{Z}_N^* \]
\[ A = k^e \mod N \]
\[ B = m \times (k + 1)^e \mod N \]

Then, \(C = (A, B)\).

---

**Decryption of** \(C = (A, B)\)

\[ k = A^d \mod N \]
\[ m = B / (k + 1)^e \mod N \]

---

**Fig. 1.** The DRSA Encryption Scheme

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 \(m \in \mathbb{Z}_N^*\) to Alice whose public key is \((N, e)\), Bob chooses a random \(k \in \mathbb{Z}_N^*\) and computes

\[ A = k^e \mod N \]
\[ B = m \times (k + 1)^e \mod N \]

He sends the pair \((A, B)\) to Alice. When she receives a pair \((A, B)\), Alice computes

\[ k = A^d \mod N \]
\[ m = B / (k + 1)^e \mod N \]

4.2 Security Properties

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

**Theorem 9.** The DRSA encryption scheme is semantically secure against chosen-plaintext attacks relative to the Decisional Dependent–RSA problem.

**Proof.** Let us consider a set of messages \(A_1, A_2\) and corresponding encryptions \(C_1, C_2\). If the DRSA encryption scheme is semantically secure, then the attacker should not be able to distinguish between these two sets of ciphertexts. Let us consider a set of messages \(A_1, A_2\) and corresponding encryptions \(C_1, C_2\). If the DRSA encryption scheme is semantically secure, then the attacker should not be able to distinguish between these two sets of ciphertexts. Let us consider a set of messages \(A_1, A_2\) and corresponding encryptions \(C_1, C_2\). If the DRSA encryption scheme is semantically secure, then the attacker should not be able to distinguish between these two sets of ciphertexts.
In the figure beside, we construct a D-DRSA adversary, $B$, 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.

---

$\textbf{B} \alpha, \gamma$

- Run $A_1(pk)$
- Get $m_0, m_1, s$
- Randomly choose $b \in \{0, 1\}$
- $A = \alpha$, $B = m_b \cdot \gamma \mod N$
- 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 $(\alpha, \gamma)$ comes from the random distribution. But in this case, one can see that the pair $(A, B) \in \{r, m_b \cdot s \mod N \mid r, s \in \mathbb{Z}_N\}$ is uniformly distributed in the product space $\mathbb{Z}_N^* \times \mathbb{Z}_N^*$, hence independently of $b$. Then $\Pr_{\text{Rand}} B \alpha, \gamma \quad \Pr_{\text{Rand}} c \ b \ = \ 1$.

On the other hand, when the pair $(\alpha, \gamma)$ comes from the DRSA distribution, one can remark that $(A, B)$ is a valid ciphertext of $m_b$, following a uniform distribution among the possible ciphertexts. Then $\Pr_{\text{DRSA}} B \alpha, \gamma \quad \Pr_{\text{DRSA}} c \ b \quad c \ b \ \text{def} = 1/2 \pm \text{Adv}_A$.

The advantage of $B$ in distinguishing the DRSA and the Rand distributions is $\text{Adv}_B \text{Adv}_A/2$, and therefore greater than $\varepsilon/2$.

\[\square\]

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 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 $\ell$-bit numbers. The initialization is unchanged. To encrypt a message $m \in \mathbb{Z}_N^*$ to Alice whose public key is $(N, e)$, Bob chooses a random $k \in \mathbb{Z}_N^*$ and computes $A = k^e \mod N$ as well as $B = m \times (k + 1)^e \mod N$ 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 $k = A^d \mod N$ and recovers the probable plaintext $m = B / ((k + 1)^e \mod N)$. She then checks whether they both satisfy the control padding $H = h(m, k)$.
Initialization

\( ℓ \), security parameter
\( N = pq \), a large RSA modulus
\( e \), an exponent, relatively prime to \( ϕ(N) \)

Public key: \((N, e)\)

Secret key: \( d = e^{-1} \mod ϕ(N) \)

\( h : \mathbb{Z}_N × \mathbb{Z}_N \to \{0, 1\}^f \), a hash function

Encryption of \( m \in \mathbb{Z}_N^× \)

\( k \in \mathbb{Z}_N^× \)
\( A = k^e \mod N \)
\( B = m \times (k + 1)^e \mod N \)
\( H = h(m, k) \in \{0, 1\}^f \)

Then, \( C = (A, B, H) \)

Decryption of \( C = (A, B, H) \)

\( k = A^d \mod N \)
\( m = B/(k + 1)^e \mod N \)
\( H' = h(m, k) \)

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

5.2 Security Properties

Concerning this scheme the following result:

**Theorem 10.** 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 particularly for the challenge ciphertext, which is the triple \((A, B, H)\) where \(H\) is randomly chosen in \(\{0, 1\}^f\). 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 \(k^e \equiv A \mod N\), then we stop the game, and whether \(γ = (k + 1)^e \mod N\) 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:

1. \(H'\) 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 \(A' = k^e \mod N\) and \(B' = m \times (k + 1)^e \mod N\). Then it returns \(m\) as the decryption of the triple \((A', B', H')\).
2. 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 encryptions. 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 \((\alpha, \gamma)\) comes from the \(\text{DRSA}\) distribution, since the probability of success can be improved if the adversary guesses the \(e\)-th root of \(\alpha\), which had led to stopping the game with an answer 1,

\[
\Pr_{\text{DRSA}} B(\alpha, \gamma) = 1 \geq 1/2 + \frac{\text{Adv}^A_2}{q_d^2 \ell},
\]

where the adversary asks at most \(q_d\) queries to the decryption oracle. However, if the pair \((\alpha, \gamma)\) 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 \(\alpha\), but then, \(B\) likely outputs 0:

\[
\Pr_{\text{Rand}} B(\alpha, \gamma) = 1 \leq 1/2 - \Pr(\alpha^d \text{ guessed} \leq 1).
\]

Therefore, \(\text{Adv} B \geq \frac{\text{Adv}^A}{q_d} - \frac{q_d}{\ell} \).

5.3 Description of the Second Variant: \(\text{DRSA-2}\)

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 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 \(k \in \mathbb{Z}_N^{*}\) and computes

\[
A = k^e \pmod{N}.
\]

He can then mask the message in

\[
B = m \oplus h_1((k+1)^e \pmod{N}),
\]

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 \(k = A^d \pmod{N}\). She can therefore recover the probable plaintext

\[
m = B \oplus h_1((k+1)^e \pmod{N}).
\]

Then, she checks whether they both satisfy the control padding,

\[
H = h_2(m, k) .
\]

Theorem 11. \(\text{The DRSA-2 encryption scheme is semantically secure against adaptive chosen-ciphertext attacks relative to the Dependant-RSA problem in the random oracle model.}\)

Proof. He resu...
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
- $h_2: \{0, 1\}^{k_2} \times \mathbb{Z}_N \to \{0, 1\}^{k_2}$, a hash function

Public key: $(N, e)$

Secret key: $d = e^{-1} \mod \varphi(N)$

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

- $k \in \mathbb{Z}_N$
- $A = k^e \mod N$
- $B = m \oplus h_1((k + 1)^e \mod N)$
- $H = h_2(m, k)$
- $C = (A, B, H)$

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

- $k = A^d \mod N$
- $m = B \oplus h_1((k + 1)^e \mod N)$
- $H = h_2(m, k)$

More precisely, let $A, A_1, A_2 \in \mathbb{Z}_N$ 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 $\varepsilon$. 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,

$$\Pr[A_2(s, m_0, m_1, E(m_b)) = b] = \frac{1}{2} \pm \text{Adv}^A,$$

then

$$\text{Adv}^A \geq \Pr[E_1 \lor E_2 \land \text{no incorrect decryption}] \geq \text{Adv}^A - q_d \times -k_2.$$

For the reduction, one just has to choose the query which should correspond to $a \mod N$ and propose a challenge guess $\oplus$ it is good choice or $b \mod N$ happens upon the other case, since the worst case is

$$\Pr[E_1 \lor E_2] \times / \Pr[A_2(b \land E_1 \lor E_2)] = \text{Adv}^A.$$
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 \( \mathbb{Z}_p^* \) of order \( q \). Each user chooses a secret key \( x \) in \( \mathbb{Z}_q^* \) and publishes \( y = g^x \mod p \).

To encrypt a message \( m \), one has to choose a random element \( k \) in \( \mathbb{Z}_q^* \) and sends the pair \(( r = g^k \mod p, s = m \times y^k \mod p)\) as the ciphertext. The recipient can recover the message from a pair \(( r, s)\) since \( m = s / r \mod p \), 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 \( d = e^{-1} \mod \varphi(N) \).

To encrypt a message \( m \), one just has to send the string \( c = m^e \mod N \). To recover the plaintext, the recipient computes \( c^d \mod N \).

**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 \( A = (m \parallel 0)^k \oplus g(r) \) and \( B = r \oplus h(A) \) and finally sends \( C = (A \parallel B)^e \mod N \). The recipient can recover the message from \( C \) first computing \( A \parallel B \oplus c^d \mod N \), then \( r = B \oplus h(A) \) and \( M = A \oplus g(r) \). If \( M \) ends with \( k \) 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.** The schemes allow precomputations. Indeed, a user can precompute many pairs for a given recipient, i.e., \((a^e \mod N, (a+1)^e \mod N)\). 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 \( p = 2^q + 1 \)), 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 logaritm-based ones, we consider 1024-bit RSA-moduli: \( n = |N| = 1024 \), \( e = 65537 = 2^{16} + 1 \), and further \( k_1 = 64 \) 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.

<table>
<thead>
<tr>
<th>Schemes</th>
<th>RSA</th>
<th>OAEP</th>
<th>El Gamal</th>
<th>DRSA</th>
<th>DRSA-1</th>
<th>DRSA-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encryption</td>
<td>RSA</td>
<td>RSA</td>
<td>RSA*</td>
<td>C-DRSA</td>
<td>C-DRSA</td>
<td>C-DRSA</td>
</tr>
<tr>
<td>Security</td>
<td>RSA</td>
<td>RSA</td>
<td>D-DH</td>
<td>D-DRSA</td>
<td>C-DRSA*</td>
<td>C-DRSA*</td>
</tr>
<tr>
<td>Plaintext</td>
<td>1024</td>
<td>448</td>
<td>511</td>
<td>1024</td>
<td>1024</td>
<td>1024</td>
</tr>
<tr>
<td>Ciphertext</td>
<td>1024</td>
<td>1024</td>
<td>2048</td>
<td>2208</td>
<td>2208</td>
<td>3232</td>
</tr>
<tr>
<td>Expansion</td>
<td>1</td>
<td>2.3</td>
<td>2</td>
<td>2</td>
<td>2.2</td>
<td>2.2</td>
</tr>
</tbody>
</table>

**Fig. 4.** Efficiency of Encryptions and Decryptions

**Remark 12.** 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 CCA-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

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 the RSA problem. Indeed, a smaller exponent, such as \( e = 65537 \) (or even 3), can be used, hence an improved efficiency is obtained:

\[
\frac{k_1}{|N|} = 1024
\]

This scheme is 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 needed for the DRSA cryptosystem to support secure encryption and decryption. Further more, 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**