Identification Based Encryption with RSA-OAEP Using SEM and Without

Rkia Aouinatou[1], Mostafa *Belkasmi*[2]
[1] *UFR SYSCOM*, Faculty of Sciences, Mohamed V-Agdal B.P. 1014 Rabat, Morocco
∗ *Laboratoire de recherche Informatique et Telecommunication: LRIT*
Email: rkiaaouinatou@yahoo.fr
[2] ENSIAS: University Mohammed V- Souissi, Rabat, Morocco
Email: belkasmi@ensias.ma

Abstract

In this article we show how we can integrate the RSA (RSA-OAEP) into the IBE. Our prove can be make with either Standard Model or Random Oracle. We firstly develop the basic ideas made in this direction, so that to create a novel scheme with which we can signs and crypt at the same time. Then we give our new approach which conserves properly the syntax of the RSA classic. Additionally we compare our authentication with the signature of Shamir. More than that, in the RSA-IBE there is the problem of relating the exponent with an identity. Even if, there was some proposals in this direction, but they operate only with the Random Oracle. And in this article we will response to question of Xuhua Ding and Gene Tsudik, in order to propose an efficient exponent for an RSA-IBE. In the end of the article we give a useful appendix.

Keywords

IBE, mRSA, SEM, RSA-IBE, Classic RSA, OAEP, CPA, CCA2, authentication, Shamir signature....

1 Introduction

IBE (**I**dentification-**B**ased **E**ncryption) is a public key cryptosystem where the public key can be represented as an arbitrary string such as an email address. The concept was proposed in 1984 by Adi Shamir without created any cryptosystem that can work with this technique. But, he applied his idea on the scheme of signature by integrating the most famous crytosystem RSA. However, integrating the RSA into the IBE is open until present, because it suffer from a lot of problems, we can cite :

1. The problem of relating the Exponent e or the Modulus n with an identity.
2. The problem that if n is not related to the identity. It will be difficult to precise, if it is for the concerned receiver. Because, IBE is an off-line to this latter.

These problems are the obstacle to the inventor of IBE and the RSA Shamir, who declared definitively that[1] it is impossible to combine this later with the first :
$\prec\prec$ Unfortunately, the RSA scheme cannot be used in a way that satisfies these conditions simultaneously :

1. If the modulus n is a pseudo-random function of the user's identity, even the key generation center cannot factor this n and cannot compute the decryption exponent d from the encryption exponent e.
2. If the modulus n is universal and the seed is its factorization, then anyone who knows an encryption exponent e and its corresponding decryption exponent d can compute the seed.$\succ\succ$

Therefore for these reasons Shamir did not integrate the RSA into the IBE. He used it only on the scheme of signature, because this last is only to authenticate the users. More than that, we will choose it (the signature) after choosing the appropriate cryptosystem and thus after setting the most appropriate publics parameters.

After the failed of Shamir, work in IBE operate only with the pairing (the first proposal is with Boneh and Franklin[2]). As a consequence, IBE was limited in their practical with the traditional public-key cryptosystems like RSA. So, it can't benefits from : twenty years of mathematical research, operational familiarity, experience and comfort with the security of this established system.
But, there was some typical techniques which essayed to manage the RSA with IBE using what is called SEM. These methods are proposed by the cryptographers : Dan Boneh, Xuhua Ding and Gene Tsudik. None of them are satisfactory, as it doesn't conserve the syntax of the RSA classic. In[5][6] the authors used a modulus fix. But, to share the same modulus by multiple users in the IBE using a normal RSA, will be utterly insecure. Because, it will be fragile to a trivial attack which can simply factor the modulus. To resist to this, the authors used the technique of SEM (SEcurity Mediator)[4], at which they divide the secret key between the SEM and user. Using the fact that none of these later can calculated the modulus n, but this make communication very low. By contrast, in our approach we can use modulus fix and we can conserve the RSA classic.
Organization : Firstly we will describe more clearly the idea of some techniques proposed : that's of Dan Boneh, Xuhua Ding and Gene Tsudik. In the second and third version we repaire this proposal, more we propose a new cryptosystem which we study its security. In fourth section we study the rigidity of our authentication against CCA and in the fifth section we compare this authentication with the signature of Shamir. At the end we gave a brief conclusion and an Appendix.

2 Some Preliminaries

2.1 Functionality of the IBE :

An IBE system contains four basic components in its construction :
Setup : A trusted central authority manages the parameters with which keys are created. This authority is called the Private Key Generator or PKG. The PKG takes a security parameter k and returns **params** (system parameters) and **master-key**. The system parameters will be publicly known, while the master-key will be known only to the (PKG).
Extract : Takes as input **params**, **master-key**, and an arbitrary ID_R, it returns a private key d_{ID_R}.
Encryption : When Alice wishes to encrypt a message to Bob, he encrypts the message to him by computing or obtaining the public key, and then encrypting a plaintext message M with params, ID_{Bob} to obtain ciphertext C.
Decryption : When Bob has C, he contact the PKG to obtain the private key S_{Bob}, he decrypts C to obtain the plaintext message M.

2.2 D-RSA- SEP

Definition 1 :[13]

(Decision RSA Short Encrypted-Prime Problem (D-RSA- SEP)). Let (l_e, l_n, l_k) be security parameters. The challenger is given a triple (e', n', z'), where n' is an l_n-bit RSA modulus, e' is an l_e-bit random prime, and $z' \longleftarrow {Z_{n'}}^*$. Its goal is to decide whether or not z' is of the form $k^{e'} \bmod n'$, for some k $\in$ Primes$(2^{l_k-1}, 2^{l_k})$.

3 Idea of Dan Boneh, Xuhua Ding and Gene Tsudik[5]

Identification Based Encryption is one of the technical offers to simplify the problem of revocation of the keys in the PKI. As we identify each person only by his identity without according him any certificate. This technique has been proposed by Adi Shamir in 1984 without implementation of any cryptosystem. And we will wait until 2001, at which Dan Boneh and Matthew Franklin[2] have proposed the first integrated and the first implementation of a cryptosystem of IBE using **Pairings**.
The inventor of IBE Shamir operates his idea by integrating the most famous cryptosystem in the world : RSA, but only on the scheme of signature.
Shamir cites two reasons for which he hasn't incorporated the RSA in the IBE, and since 1984 up to present

the integration of RSA in the IBE is almost infeasible. But there was some work which uses some typical technique to realize this trick. The method that we spirited to talk is about that of Dan Boneh, Xuhua Ding and Gene Tsudik in their article[5], in which they use what we called SEM
The SEM[4] are used to simplify the revocation of the keys, by installing in the middle a trust entity which simplifies the validation of digital signatures and the revocation of the keys, and this for any entity which need the security.

3.1 Introduction to the SEM

The SEM (SEcuriy Mediator) is a trusted entity installed in the middle to generate and to simplify the derivation of the private key.
To signs or to decrypt a message, Bob must communicate with the SEM to obtain what we called **Token** which is a symbol. Without this last, Bob can't have any private key to decrypt his message and if Bob has cheated in his communication, the administrator must confirm the SEM to stop sending the token to the public keys of Bob
Among the most important services of the SEM is the validation of the signatures every time we need it, by returning the certificate which is more desirable in comparison if we have communicated directly to the CA. Thus the presence of the SEM is invisible to any user not interested in the key in question.
The SEM are integrated in the communication of the RSA to simplify the revocations of the keys. The cryptosystem we should mention is : mRSA (ie RSA with mediate SEM) works with this technique.

3.2 Mediate RSA : mRSA

mRSA is a convenient method to simplify the communication between the user and the CA in the cryptosystem RSA, to obtain secure private key by the intermediate of the SEM. However the communication require an understanding and a cooperation between user and the SEM without cheating each entity the other.
The main idea of mRSA is that the derivation of the private key is divided in two parts, one for SEM and the other for the user. So, without cooperation between the two entities, private key remains unknown (for more detail see[4])
Dan Boneh, Xuhua Ding and Gene Tsudik incorporated mRSA in the IBE basing on the SEM[5]. For realizing this, these authors used +mRSA (the + mRSA has been invented by Bonneh et al[5]. As canesan has proposed another version mRSA*) and during the rest we are only interested in +mRSA because it is the version used by Dan Boneh, Xuhua Ding and Gene Tsudik

3.3 IB-mRSA ie IBE with mRSA

3.3.1 Description of the Method

Problem to relate e with ID

In the initialize phase the trust authority PKG publishes their parameters : n = $p \times q$ fix and specialized to the number of user which are in the same organization, hash functions, a useful parameters for one-way function in a secure site (specialized to an organization or domain). The concerned user extracts these parameters according to a certificate.
So Alice who has the identity of Bob : ID_B wish to send him the message $M_{Alice}^{e_B}$ mod n. Firstly she consults the site published by the PKG in which Bob is a member according to his Domain (for more detail see[5]) and she extracts the appropriate parameters but before she calculates the encryption. Alice needs to calculate an e_B corresponding to the ID_B and to calculate it we propose tree methods :
The first is that of[5]. According to this, $e_B = f(ID_B) = 0^{k'_0}||ID_B||00000001$ with
$k'_0 \longleftarrow k - |ID_B| - 8$, k is the secret parameter (f must be one-way function ie we can't have a coincidence on identity.) □
The second one is that of[6]. In[6] $e_B = 0^s||KG(ID_B)||1$ with KG is a hash function such as MD5 or SHA1 and s is the secret parameter selected by the PKG. We privilege this choice because according to it the authors prove that the scheme is CCA2 secure taking into account that they conserve the same scheme of[5]. The only change they make is in the formulation of e_B □
The third one is **Our**. In our proposition we have taken into consideration the perspective proposed in[6].

However, Xuhua Ding and Gene Tsudik announce that they need to investigate alternative mapping functions that can produce more "efficient" RSA exponents. And it may be preferable to choose it in standard Model. In our proposition the e_B is as follow :
Firstly our goal is to choose an e_B (generally e_{ID}) in the standard Model and which verify the condition of RSA i.e e_B prime with the Euler function φ(n)=$(p-1)\times(q-1)$. We propose to construct an e_B **prime** and of length approximately $\frac{3}{4}$ the length of the module n. Effectively e_B which is surely greater than (p-1) and (q-1) will be prime with their product (theorem of Bezout).
In the first pad we will project all ID in the code ASCII. This projection will be different from each other because all ID are different(if not the term identity has any sense). And we multiply it by an odd prime of length 2^k (for example 2^k+1), k is an appropriate parameter preceded by the PKG and fix for all ID. For example, if our n is of 2048-bit, according to our procedure we wish to search for an e_{ID} of length $\frac{3}{4}$.2048= 1536 bit. As the ID may have for example at most 35 characters taking into account the characters of the organization (for example : @Journal of Secuity.com) which we can eliminate it (since the organization is the same for their users). However, our ID may be of 15 characters i.e of 120-bit. So, the k that can be published by the PKG is of 1410 bit ((1536-120-6)bit, the last 6 bits are reserved to the search of an prime integer). The projection of ID in ASCII which we take as number (for example if the projection is in form 011011 our number is 11011). If this number is odd i.e it has 1 in his least bit we take it, if not we change this least 0 bit by 1. We multiply this number for example by $2^{1410}+1$ and we test if this number is prime or not using the Appendix or the theorem of Wilson (even if it is expansive for large prime) :
Theorem of Wilson : An integer p strictly superior to 1, is a prime number if and only if it divide (p - 1) ! + 1, more precisely if and only if : (p - 1) ! + 1 = 0 mod p. □
But if we get a number not prime, we add to it 2 until we get a prime number (the prime numbers are dense in $\aleph$: set of integer). For more detail see the Appendix
We have constructed a **prime** number greater than (p-1) and (q-1) so it will be prime with (p-1), (q-1) and so with their product (p-1)×(q-1). As e_{ID} constructed is prime it is rigid against the attack of non-malleability. More of that all the e_{ID} constructed are different from one another (all the ID are different + multiplication by fix number 2^k+prime numbers are dense).
This proposition is useful, because we will have an e_{ID} standard as the RSA classic and we will not have the problem of malleability compared to[6]. That's we will see in the sequal.
The only weakness of this method is that those e_{ID} has a long term which may show down the calculation. But the length of our parameter is small compared to[6] which has length equal to that of n □

Simple variation in the idea of Boneh, Xuhua and Tsudik

In the idea[5] we add a new approach, especially in the system of communication, since[5] may be expensive. Because, to distribute d_u we need a secure channel also we have the problem that the user will authenticate to the SEM as it authenticate to the PKG. This authentication is very low because it's in the traditional form. By contrast, in our method the SEM will only declare and the authority may only control.
We propose that Bob, who has no private key to decrypt the message send by Alice $M_{Alice}^{e_B}$. Choose a secret key d_{u_B} and send $d_{u_B}^{e_B}$ to the authority in the following way :

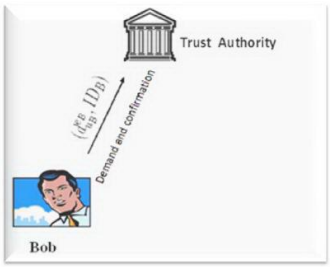

FIG. 1 – Demand and Authentication

And it is simple to the authority to extract the key d_{u_B} from $d_{u_B}^{e_B}$ mod n, because it is the only who know p,q.
The aim of this shipment is to confirm the authority (PKG) this is my chosen private key, and this is my

Identity, I need complete key to communicate. So the authority takes into consideration this demand, she stocked somewhere (in a record) the couple (d_{u_B}, ID_B), and she waits for the second request.
Bob makes a contact for the second time the PKG (message $(H(m), H(m)^{d_{u_B}})$, with m is an arbitrary message) but in this faith by the intermediate of SEM in the following way :

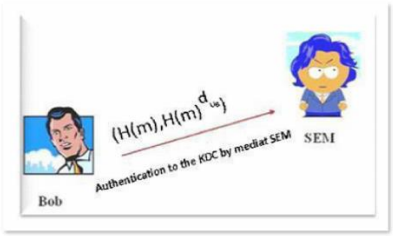

FIG. 2 – Second contact : party 1

The aim of this tie is to allow SEM to authenticate Bob to the PKG.
So the SEM say to the PKG this is a person who asks the key d_{SEM} you can check its identity. The PKG consult her record to verify the selected key by Bob and after she calculates :

$d_{SEM} = d_B - d_{u_B}$
$H(m)^{d_{SEM}}$
$H(m)^{d_{u_B}} \times H(m)^{d_{SEM}}$
H'(m)=$(H(m)^{d_{u_B}} \times H(m)^{d_{SEM}})^{e_B}$

After verifying the identification of Bob (H '(m) = H (m))
The PKG sent d_{SEM} to SEM but this must be across the secure channel (since the SEM is a drift of the PKG)

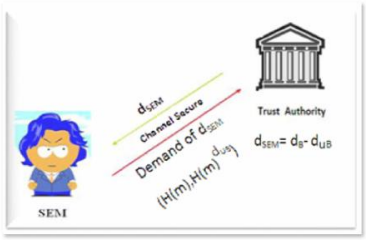

FIG. 3 – Second contact : party 2

The SEM should be authenticated to Bob (message $H(m), H(m)^{d_{SEM}}$)), to tell him that your application is accepted. So Bob who verifies more the identification of SEM (we need this verification to not broadcast the message to anyone), sent the message that is sent by Alice (${M_{Alice}}^{e_B}$ coupling with $H(m)^{d_{u_B}}$ i.e signature) to the SEM to obtain a Token (help). Then the SEM who received d_{SEM} sent to Bob $({M_{Alice}}^{e_B})^{d_{SEM}}$, this is clearly seen in the figure 4 :

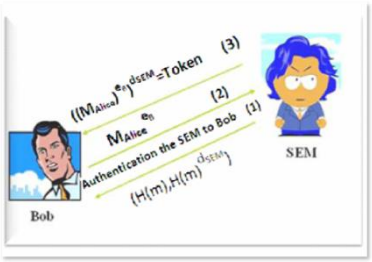

FIG. 4 – Authentication+Token

In the last Bob computes :
$({M_{Alice}}^{e_B})^{d_{SEM}} \times ({M_{Alice}}^{e_B})^{d_{u_B}}$ mod n= $({M_{Alice}}^{e_B})^{d_B} mod n = M_{Alice}$

The algorithms used to realize these ties (calculation, verification ...) are summarized in[5]

3.3.2 Security

The security of this method is first linked to the security of RSA-OAEP (see section 4.1.1). Secondly it is impossible to attack the key d_B unless we attack the SEM (of course the SEM is a trusted entity). Because even if we attack d_{u_B} it remains the second slice d_{SEM} which rest to the SEM, since this latter had obtained this key with a high security (secure channel)
So the security of this cryptosystem is linked to :

The security of RSA-OAEP
The understanding and fidelity of the SEM.

For more detail see[5].
Advantage :
With this method we gain to utilize a fixed n and integrate without problem the RSA in the IBE (which is our goal)
Drawbacks :

* An Authority who knows all our secrets (the other two entities SEM and Bob are unable to extract the complete key because each of them had only half key)
* Need for each message to the token and instead of a single contact (Authority) we need at least two (Authority + SEM).
* Problem of the discrete logarithm (PLD) because if we choose d_{u_B} big, so d_{SEM} may be small (or reverse), therefore it may be vulnerable to PDL (this weakness is related to our procedure)

4 Our Proposal

4.1 Problematic :

Assuming that we have secrets and we want none person to know it, even the PKG who is a trusted authority. For example, it may be the military secrets or business secrets. That will really rise a problem.
We will present a cryptosystem in which we will repaired the precedent weakness (dependence on the PKG) and in which we reduce the contact to the SEM (partially or definitely) and also reduce the contact to the PKG.

3.1.1 First Version

3.1.1.1 Half Independence

Setup :

Select a security setting n = $p_B \times q_B$ as p_B and q_B are two very large prime numbers (n may be of 3072 bit taking into account the current security).
The parameters of publications are : $\prec n, H, Z_n \succ$, with H is Hash Function (preferably SHA1). It is recommended to obtain these parameters with a certificate.
We calculate e_{ID} as previously e_{ID} (using our method)

Extract :

Assuming that Alice who knows Bob's identity and therefore his public key, sends him a message. Then Bob which hasn't private key, calculates d_{u_B} he sent it to the PKG. After that he calculates $(H(m_B), H(m_B)^{d_{u_B}})$ (the m_B is a message chosen by Bob), and he sends it to the SEM.

Encrypt :

Alice chooses $S_{Alice} \in Z_n$.
Alice (recipient) calculates $((M_{Alice} + S_{Alice})^{e_B}$ mod n,$S_{Alice}^{e_B}$ mod n) = (u,v)

Decrypt :
After receiving $(H(m_B), H(m_B)^{d_{SEM}})$ from the SEM and after he is authenticate to the SEM as in the previous method. Bob sends (u, v) to the SEM
The SEM responds by sending ($u^{d_{SEM}}$ mod n,$v^{d_{SEM}}$ mod n)=(u',v') to Bob. Having received this answer, Bob computes $v'^{d_{u_B}}$ mod n = S_{Alice}, and
$u'^{d_{u_B}}$ mod n = $M_{Alice} + S_{Alice}$. In the end he calculates the difference $v'^{d_{u_B}}$ mod n-$u'^{d_{u_B}}$ mod n = $M_{Alice} + S_{Alice}$- S_{Alice}=M_{Alice}
The S_{Alice} chosen by Alice must be fixed throughout the communication Alice-Bob. By S_{Alice} Bob authenticate Alice every time he communicates with her, because if we assume that an adversary Eve wants to know the content of the communication (Alice-Bob) she sends a message M_{Eve} according her precautions of the contents of the communication to Bob.
But, since Eve can not access to S_{Alice} she has chosen S_{Eve} and she sends
$((M_{Eve} + S_{Eve})^{e_B} mod n$,$S^{e_B}_{Alice}$ mod n). And Bob must find that
$(M_{Eve} + S_{Eve})^{e_B} - S^{e_B}_{Alice}$ is a message incoherent, since $S_{Eve} \neq S_{Alice}$.
If we want a full authentication, our procedure is as follows :
The first message sent by Alice must be of the form $(M_{Alice})^{e_B}$, after she sends the same message in the form : $((M_{Alice} + S_{Alice})^{e_B}$ mod n , $S^{e_B}_{Alice}$ mod n)
Using the same first message M_{Alice}, it is to say to Bob, I am the same user and this is my proposed modulus.
In all the communication, all the messages will be sent as follows : $((M'_{Alice} + S_{Alice})^{e_B} mod n, S^{e_B}_{Alice}$ mod n)
This is for message (Alice-Bob), for the reverse (Bob-Alice), we wish Alice to not contact any center of security to obtain her private key (half independence). To do it, so we must do as follows :
The S_{Alice} chosen by Alice must be as follow $S_{Alice} = p_{Alice} \times q_{Alice}$ and not factorials : product of two large prime numbers in Z_n (the S_{Alice} for example is of 2048-bit)
So after having received the message by Bob. To respond to this message, Bob must response in the following way :
He has chosen firstly S_{Bob} ($*$) in Z_n. He sent his message according to the module S_{Alice} instead of n : $((M_{R_{Bob}} + S_{Bob})^{e_A}$ mod S_{Alice}, $S^{e_A}_{Bob}$ mod S_{Alice}). The M_{Bob} is the response of Bob.
To decrypt this message Alice has already calculated d_A such that :
$d_A \times e_A$=1 mod $\varphi(S_{Alice})$. So she is capable to decrypt the message.
The d_A is only known by Alice because she is the only one who knows p_{Alice},q_{Alice}.
We summarize all this in the figure 5 :

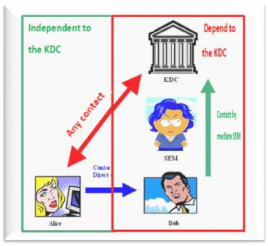

FIG. 5 – Half Independence (Version : 1)

The figure 5 is divided into two parts, one that depends on the SEM and PKG (at the right, framed in red) and the other not (at the left, framed in green)
We want to make the independence to the party framed in red i.e at the right.

3.1.1.2 Total Independence

If the S_{Bob} $(*)$ chosen by Bob in the previous subsection has the form : $S_{Bob} = p_{Bob} \times q_{Bob}$ and not factorial (the p_{Bob} , q_{Bob} are two large prime numbers only known by Bob).
So Alice must send her messages in the form :
$((M_{Bob} + S_{Alice})^{e_B}$ mod S_{Bob},$S^{e_B}_{Alice}$ mod S_{Bob}).
The PKG is unable to read the message sent by Alice, since it can not factorize S_{Bob}. So after this, Bob may

communicate directly with Alice without need to any Token nor contact with the SEM.
The problem with this technique is that the opponents have an immense freedom to contact Bob instead of Alice. So Bob can receive a very large number of messages without value and he can not clearly authenticate his sender. More than that Bob may have a problem of revocation of the S_{ID_i}. To limit this we propose a second version in the next section

3.1.2 Second Version

In the figure below [Figure 6] we assume that the entities Alice and Bob during their first messages contact the PKG by the intermediate of the SEM. And after a certain moment they agree to utilize their chosen parameters (principle of cryptography hybrid, but in this faith we do not spend a symmetrical clef, we spend only the public parameters : S_{Alice}=$p_{Alice} \times q_{Alice}$ and $S_{Bob} = p_{Bob} \times q_{Bob}$ not factorials for t $\leq t_1$. The p_{Alice}, q_{Alice} and p_{Bob},q_{Bob} are only known respectively by Alice and Bob). So after t $> t_1$ Alice and Bob are independent from the PKG

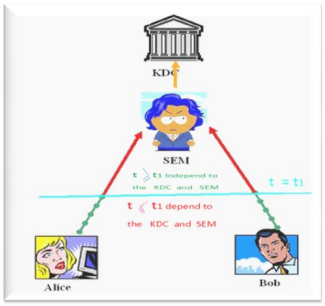

FIG. 6 – Independence Total (Version : 2)

There may be other versions, but we must say that the first receptor of the message (first message emitted during the communication) is unable to release it without the help of the SEM.
The security of these two versions is related to the security of the RSA classic and to the understanding and the trusting during the communication (fidelity of the SEM, integrity of data and identification).
The problem for these two versions is that our chosen parameters can be changed. But this can be resolved easily, since we get these parameters through the PKG. So if there is a change (the change can be easily seen if we can not decrypt the message because of the change of the module S) only the PKG is able to do, and we can settle this with him.
Also in these two version we still have the PLD i.e the third weakness. To solve this problem we propose that the authority publish for example their module n in 8192-bit and so any receiver must not repass 4096 bit in his choice of the private key ($2048 < d_{u_r} < 4096$)
NB : For changing the parameter S_{Alice} and S_{Bob} (problem of revocation). Alice and Bob can return to the method of the SEM (only for change).
In both these versions 1 and 2 we have used typical techniques. But it is the custom of thirty years that the construction of RSA is related to the factorization. In the next version we return to our RSA related to the factorization.

4.2 Third Version

In this paragraph we utilize the following arguments in addition to the bearing of what we called OAEP as in the previous section

- □ RSA **strong**. An RSA is said to be strong if it has the strong **prime** number. The number prime p is said to be strong if $\frac{p-1}{2}$ is prime[7].
- □ Idea of the RSA **multiprime**[8].

Contrary to what we have already stated : contact the PKG with the intermediate of the SEM. In this section we can't utilize this technique and we propose the following method :

The KDC publish a fixed n, this n is in the form : n=(2p'+1)(2q'+1), p 'and q' are two strong prime (principle of RSA strong) and with for example 2048 - bit (so n is of 4098-bit).
For each receiver who demands private key d_r, the PKG try to calculate this later. It seems that the PKG computes d_r and simply it sends it to the concerned receiver, but how the PKG responds all the ID with fixed n ? !
To resolve this we propose in addition to the generation of n, the PKG select for each demand k the primes numbers $p_{i_{r_k}}$ and $q_{j_{r_k}}$ such that : $e_{ID_{r_k}} \times d_{r_k} = 1 \bmod (p_{i_{r_k}} \times q_{j_{r_k}} \times \varphi(n))$
The e_{ID_r} is calculated as previously

Concreteness :

After the Fermat theorem we have :
$\begin{cases} M^{p-1} = 1 mod p \\ M^{q-1} = 1 mod q \end{cases}$.
Using the Chinese remainder (reste chinoix) we have :
$M^{(p-1)(q-1)} = 1$ mod n
So $M^{p_{j_{r_k}} \cdot q_{j_{r_k}} \cdot (p-1).(q-1)} = (1 mod n)^{p_{j_{r_k}} \cdot q_{j_{r_k}}} = 1$ mod n
And $M^{e_{ID_{r_k}} \times d_{r_k}} = M^{1+k' \times p_{i_{r_k}} \times q_{j_{r_k}} \times \varphi(n)}$=M.(1 mod n)= M mod n

4.2.1 Method of generation with a security analysis

Alice who wishes to send a message to Bob contacts for the first time an authority (PKG) to extract the parameters n (which is fix) and the appropriate parameter to generate e_{Bob} (as previously). When the user Bob has received this message from Alice encrypted with his identity, he must contact the PKG to obtain the private key d_B.
After calculating $\varphi(n) = 4 \times p' \times q'$ by the PKG, for example for 5 asks, in Extract the PKG chooses five primes p_1, p_2, p_3, p_4, p_5 with 2048 - bit for example such that :
For each e_{ID_i}, i $\in \{1, 2, 3, 4, 5\}$ (Bob identity may be one of the five). The PKG computes d_{ID_i} which correspond to $\varphi'_i(n)$. We reserve this later to the corresponding recipient as follows :
$e_{ID_i} \times d_{ID_i} = 1$ mod $(p_i \times p_{i+1} \times \varphi(n)) = 1$ mod $\varphi'_i(n)$ $\forall$ i $\in$ {1,2,3,4,5}. So we have :

$e_{ID_1} \times d_{ID_1} = 1$ mod $(p_1 \times p_2 \times \varphi(n)) = 1$ mod $\varphi'_1(n)$
$e_{ID_2} \times d_{ID_2} = 1$ mod $(p_2 \times p_3 \times \varphi(n)) = 1$ mod $\varphi'_2(n)$
$e_{ID_3} \times d_{ID_3} = 1$ mod $(p_3 \times p_4 \times \varphi(n)) = 1$ mod $\varphi'_3(n)$
$e_{ID_4} \times d_{ID_4} = 1$ mod $(p_4 \times p_5 \times \varphi(n)) = 1$ mod $\varphi'_4(n)$
$e_{ID_5} \times d_{ID_5} = 1$ mod $(p_5 \times p_1 \times \varphi(n)) = 1$ mod $\varphi'_5(n)$

Under this method the PKG can handle with fix n easily 5 requests (the demand of Bob is among the five). For this method or rather this choice to be rigid against adaptive choice they must verify :

1. $\forall i, \in \{1, 2, 3, 4, 5\}$ $p_i \times p_{i+1}$, $p_i \times p'$, $p_{i+1} \times p'$, $p_i \times q'$, $p_{i+1} \times q'$ must be not factorial (it is feasible because our product is of 4096-bit).
2.$\forall i$ $l(p_i) >> l(p_{i+1})$, with for example $l(p_i) - l(p_{i+1}) = 512$ bit
(l is the Length. Their difference means the weight or rather the distance of Hamming, and >> means that it exist a difference).
3.$\forall \{i, j\} \in \{1, 2, 3, 4, 5\}$ $(p_i \times p_{i+1}) + (p_j \times p_{j+1})$ contains the number $p_{i,j}$ such that :
$p_{i,j} \times p' \times q'$ must also be not factorial.
4. $\forall i \in \{1, 2, 3, 4, 5\}$ $\varphi'_i(n)$ didn't divide $\varphi'_{i+2}(n)$.

Are those choices effective ? the answer is yes.
The first condition is intended to not extracted any of pi, p', q' from the product
The second condition is designed to resist against chosen adaptive. Because for the well chosen of (k_i, k_j) in which we hope to attack $\varphi(n)$, p',q'.. (for (i,j) $\in$ {1,2,3,4,5}) our method of generation may be vulnerable to chosen adaptive. But in reality it's not, as we can see in the following cases

a) **For the i successive**
$k_i(p_i \times p_{i+1})\varphi(n) - k_{i+1}(p_{i+1} \times p_{i+2})\varphi(n) = p_{i+1}(k_i p_i - k_{i+1} p_{i+2})\varphi(n) = p_{i+1}\varphi(n)$
We can reach to this because $p_i \bigwedge p_{i+2} = 1$, but we must choose the right (p_i , p_{i+2}) and to do it, it is with

probability $\frac{(C^2_{n_p(2^{2048})})}{(2^{2048})}$ (C is the combination). Choosing the right (p_i , p_{i+2}) is linked to the condition : $\mid p_i \mid - \mid p_{i+2} \mid = \mid p_i \mid - \mid p_{i+1} \mid + \mid p_{i+1} \mid - \mid p_{i+2} \mid$ = (512 + 512) bit = 1024 bit.
So this probability is calculated as the form : P(B / A) = $\frac{P(A \bigcap B)}{P(A)} \simeq$ P(B). We may neglect the event A in comparison with B as it is simply verified. Moreover we should only take it in to consideration when we have obtained the appropriate prime p_i and p_{i+1}.
(B is the event to choose two appropriate prime numbers among the numbers of the 2^{2048} bit. Until A is the event that the two prime numbers must verify the condition 2)
But even if we reach this result i.e $p_{i+1}\varphi(n)$, for attacked $\varphi(n)$ we have the probability : $\frac{c}{n_p(2^{2048})}$. The c is the number of the choice to attack p_{i+1}
(P($\bigcup_j p_j$)=$\sum_j P(p_j)$ = c, since $P(p_j)$ which is a probability to attack p_j from the numbers prime is an equiprobable probability).
And to attacked for example $\varphi'_i(n) = p_i p_{i+1}\varphi(n)$ (knowing that we have assumed that we have attacking $p_{i+1}\varphi(n)$) we must try with probability : $\frac{c'}{n_p(2^{2048})}$ (c' is the probability to attacking p_i)
So
$P(\varphi(n)/(k_i, k_{i+1})) \simeq \frac{\frac{c}{n_p(2^{2048})} \times \frac{(C^2_{n_p(2^{2048})})}{(2^{2048})}}{\frac{(C^2_{n_p(2^{2048})})}{(2^{2048})}} = \frac{c}{n_p(2^{2048})}$ which depends on c, but since p_{i+2} is unknown this probability is negligible.
The term $P(\varphi(n)/(k_i, k_{i+1}))$ means that the event $\varphi(n)$ is calculable knowing that we chose the right (k_i, k_{i+1}), but choose it, is after a good choice of (p_i, p_{i+1}).
(Has noted that (k_i, k_{i+1}) can be calculated from φ'_i and φ'_{i+1})
As a consequence to all this we have :
$P(\varphi'_i(n)/(k_i, k_{i+1})) = P(\varphi'_i(n)/p_{i+1}\varphi(n)) \simeq \frac{\frac{c'}{n_p(2^{2048})} \times \frac{c}{n_p(2^{2048})}}{\frac{c}{n_p(2^{2048})}} = \frac{c'}{n_p(2^{2048})}$
And we may see that even if we attack $\varphi(n)$ (this can not be attacked only after a proper choice of the p_j successive according to the above method of chosen adaptive). But for attacking any of $\varphi'_i(n)$ which is more important than $\varphi(n)$, we must do the choice with probability of $\frac{n_p(2^{4096})}{2^{4096}}$ <<1 ! !.
($\frac{n_p(2^{4096})}{2^{4096}}$ is the probability of attacking the $p_j p_{j+1}$ in $\varphi'_j(n)$ after attacking $\varphi(n)$)

b) **For the i not successive**
To attack $\varphi(n)$ from $k_i(p_i \times p_{i+1})\varphi(n) - k_j(p_{j+1} \times p_{j+2})\varphi(n) = (k_i p_i \times p_{i+1} - k_j p_{j+1} \times p_{j+2})\varphi(n)$ after a good chosen of (k_i, k_j) has probability less than : $\frac{2n_p(2^{2048})}{2^{2048}}$ or $\frac{1}{2^{524288}}$.
We can show the probability $\frac{2n_p(2^{2048})}{2^{2048}}$ and we leave to the readers the occasion of demonstrating the second.
For example for $(p_1 \times p_2 - p_3 \times p_4)$ we have :
$\mid p_1 \times p_2 - p_3 \times p_4 \mid > \mid\mid p_1 \mid\mid p_2 \mid - \mid p_3 \mid\mid p_4 \mid\mid$. Since the relationship > is an order relation on $\Re$, then one of the terms $\mid p_1 \mid\mid p_2 \mid$,$\mid p_3 \mid\mid p_4 \mid$ is greater than the other. We Assume that it is
$\mid p_1 \mid \mid p_2 \mid$ hence :
$\mid\mid p_1 \mid\mid p_2 \mid - \mid p_3 \mid\mid p_4 \mid\mid = \mid p_1 \mid\mid p_2 \mid - \mid p_3 \mid\mid p_4 \mid$
So : $\mid p_1 \times p_2$ - $p_3 \times p_4 \mid > \mid p_1 \mid\mid p_2 \mid - \mid p_3 \mid\mid p_4 \mid$ and since $\mid p_1 \mid >$ - $\mid p_3 \mid$ which give that
$\mid p_1 \times p_2$ - $p_3 \times p_4 \mid > \mid p_1 \mid\mid p_2 \mid - \mid p_3 \mid\mid p_4 \mid >$ -$\mid p_3 \mid\mid p_2 \mid - \mid p_3 \mid\mid p_4 \mid > -\mid p_3 \mid (\mid p_2 \mid + \mid p_4 \mid)$. If we apply the probability P to this inequality. These events are uniform and sure, therefore :
P($\mid p_1 \times p_2 - p_3 \times p_4 \mid$) < P(-$\mid p_3 \mid (\mid p_2 \mid + \mid p_4 \mid)$) < P(-$\mid p_3 \mid$) + P($\mid p_2 \mid + \mid p_4 \mid$))
(because P(ab)≤ P(a)+P(b), $P(a \bigcap b) = P(ab) = P(a) + P(b) - P(a \bigcup b)$)
And as we said that < is the relationship of order (we are interested to compare $\mid p_2 \mid$ and $\mid p_4 \mid$)) So for example we have :
P($\mid p_1 \times p_2 - p_3 \times p_4 \mid$) < P(-$\mid p_3 \mid$) + $P(2 \mid p_2 \mid) \leq \frac{2n_p(2^{2048})}{2^{2048}}$
(calculate P(2 $\mid p_2 \mid$) is equivalent to calculate P($\mid p_2 \mid$))

The third condition is for reason to not attack $\varphi(n)$. But even if we have attack it, we must attacked $\varphi'_i(n)$ and this after attacking p_i, p_{i+1} or $p_i \times p_{i+1}$ who have respectively the probability of attacks : $\frac{n_p(2^{2048})}{2^{2048}}$ and

$\frac{n_p(2^{4096})}{2^{4096}}$. We see that the two are strictly less than 1 (i.e negligible).
The fourth condition is of the fact that : if $\varphi'_i(n)$ divide $\varphi'_{i+2}(n)$ (or inversely) then $\varphi'_i(n) = \text{cte} \times \varphi'_{i+2}(n)$ so $d_{ID_i} = cte \times d_{ID_{i+2}}$ which is a risque.
$\varphi'_i(n)$ divide $\varphi'_{i+1}(n)$ is excluded because if we have it, we may have p_i divide p_{i+2} which it does not, since p_{i+1} is the number prime.

Remarks :

1. The term φ' isn't the Indicator of Euler it is only the notation
2. We have replaced the term **l** :length by || for the mathematical reasons
3. $n_p(x)$ is the number of the number prime that exists in [2 , x] (see the Appendix)
4. The possible values for i not successive are : $(p_1 \times p_2 - p_3 \times p_4)$, $(p_1 \times p_2 - p_4 \times p_5)$, $(p_2 \times p_3 - p_4 \times p_5)$.

The $\varphi'(n)$ is the product of four primes, with the degree of security of 8192 bit. And this, is well convenient to the following table[8], taking into account the security of multiprimes RSA

Tab. 1 – Numbers primes in a given modulo of bit

modulo in bit	1024	2048	4096	8192
number of primes	3	3	4	5

According to this table for a level of 8192-bit (we can utilize 4096-bit to reduce the complexity). We can integrate 5 prime number, but our $\varphi'(n)$ is constituted only by four prime, so we can nowhere factor it.
With these five primes we have served five identities, since find a number prime of 2^{2048} bit is expensive (see the Appendix). Can we repeat these parameters ? the answer is yes, because for example with p_1 we can associate the $p_1 p_i \varphi(n)$ such that $p_1 p_i$ satisfy the condition 2. Then we have i value of φ' so that with an p_1 we may serve i identity.
As a numerical estimate we can use with this p_1 all the prime numbers p_i which existed in the interval $[2^{512}, 2^{2048} - 2^{512}]$. The boundless of this interval is linked to the effect that if we want to choose one number of multiprimes constructed of four numbers it is preferable to pass 512 bit. Until the bounds upper is based on the condition 2.
This principle also applies to other p_i(recycles of the utilization).

Contrary to what we saw earlier : attack the complete key unless we attack the SEM. Since, in this version we didn't use the SEM, so we should therefore consider the rigidity of our key.

4.2.2 Security Analysis

4.2.3 Attack against the public key e_{ID}

The encryption key can't caused the attack of the cryptosystemes even if it may have coincidence. Since it is linked to the identity and therefore it is within to the reach of everybody. So even if the Adversary Eve who construct an $e_{ID_{Eve}} = e_{ID_{Bob}}$ can't obtain anywhere d_{ID_B} as she doesn't have ID_{Bob} (the prime number changes for each identity).
We assumed that the size of e has no influence on the attack of the cryptosystem's according to[9].

4.2.4 Attack against the decryption key d_{ID}

The security of d_{ID} is linked to $\varphi'(n)$ ($e_{ID} \times d_{ID} = 1 \bmod (\varphi'_{ID}(n))$), can we attack this $\varphi'(n)$ which occurs in its expression an $\varphi(n)$ fix ?

4.2.5 Attack against $\varphi(n)$

The $\varphi'(n)$ is the product of two prime numbers and $\varphi(n)$, since the attack of $\varphi(n)$ has a probability : $P(\varphi(n)/(k_i, k_{i+1})) = \frac{c}{n_p(2^{2048})}$ which is a very low probability. Thus our φ is the product of p 'and q'. Can we extracted one of them from $\varphi'(n)$?

4.2.6 Attack cons p',q' $\ll$ Master Key $\gg$

Our RSA is **strong** then $\varphi(n) = 4 \times p' \times q'$ can not be factorial. But we contribute $\varphi(n)$ for each ID and every time we multiply it by the prime numbers which we change. Can we from several ID attacking p ', q' ? The fact of integrating 4 prime numbers in $\varphi'(n)$ leave the opponents unable to detect what part they attack. And therefore even if they attack somewhere p ', q' they can not come to define the parameters of their attack. More, those parameters are of 2048-bit !

4.2.7 Attack against $\varphi'(n)$

Our $\varphi'(n)$ is constructed from 4 prime and after all we have seen the $\varphi'(n)$ can not be attacked.

5 Security against the attacks of Simulations

Before speaking on security of our cryptosystem, we review the following reminders :
The security of a cryptographic scheme combining the possible goals and attack models. The most important goal is indistinguishability (IND) and Semantic Security. Regarding attacks, chosen-plaintext attacks (CPA) and chosen-ciphertext attacks (CCA) are the most well-known models.

Indistinguishability : Is a technical goal, aimed at capturing a strong form of privacy and being easier to work and reason with than semantic security.

Semantic Attack : The semantic security means that no information can be extracted or calculated in the average polynomial time from the ciphertext.

CPA : Is the abbreviation of Chosen Plaintext Attack ie during the studies of simulations the opponent has advantage to access to the encrypted of his chosen texts.

CCA : It is an abbreviated of Chosen Ciphertext Attack, and we divide it into two parts : CCA1 and CCA2. During CCA the adversary has advantage of access to the decrypts texts he has chosen. In the CCA2 the opponent is less limited by comparison with CCA1. We must say that the CCA2 is the most powerful among all these attacks.

Combining these goals and attacks one obtains : IND-CPA, IND-CCA1, IND-CCA2, Semantic Security-CPA, Semantic Security-CCA1, Semantic Security-CCA2. But the commonly accepted security notion (which satisfy the standardization) for public-key encryption is IND-CCA2.

5.1 Semantic Attack

Semantic security of all the three versions 1,2 and 3 is related to the security semantic of RSA.
To make these version rigid against semantic attack, we must firstly ensure that the standard RSA is rigid against these attacks.
For the RSA classic all we can say is that it is probably semantic [9]. Since we can easily extract information which we can calculate in polynomial time from the encrypted message because :
e is prime, so it is odd and we can write it in the form 2k +1, so the Jacobi symbol of
y = x^e mod n is equal to the symbol of Jacobi of x because :
$(\frac{x^e}{n}) = (\frac{x^{2k+1}}{n}) = (\frac{x^{2k+1}}{p})\ (\frac{x^{2k+1}}{q}) = (\frac{x^{2k}}{p})\ (\frac{x}{p})\ (\frac{x^{2k}}{q})\ (\frac{x}{q}) = (\frac{x}{p})\ (\frac{x}{q}) = (\frac{x}{n})$.
So we must take precautions in this sense when we use the function RSA
But this does not mean that the RSA is definitely fragile against the semantics attack. Since from the ciphertext it will be difficult to extract the parity of the clear text, because we calculate it according to a modulus.
The security of RSA is not only related to the problem of factorization or finding the inverse of the exponent e. But we also have the problem of ASCII exhaustive, since the plaintext is always in the form string of the code ASCII. We assume that we encrypt one by one all ASCII characters, raise their codes to the power e modulo n. Of course this system can break easily by a simple statistical attack. In this case, the encryption of the pattern is insecure hence the idea of what is called padding OAEP[11]
(a paiding means that we may concatenate by either 0 or 1, or any arbitrary numbers)

5.1.1 Functionality of OAEP

The OAEP has been introduced by Bellare and Rogaway in 1994. We give in the following its description : In this model the message is divided into blocks of size fix completed by zeros (see the detail in the article [RSAES-OAEP Encryption Scheme][12]).
Note now such block with M (M therefore contains some message m filled with zeros). In addition we are availing a random generator number G and a hash function H'.
We are raffling at random a number r with the common size of the germ of the random generator G and with the output of the function hash H'. $G : \{0,1\}^{k_0} \leftarrow \{0,1\}^{k-k_0}$ and $H' : \{0,1\}^{k-k_0} \leftarrow \{0,1\}^{k_0}$
Let f and g two permutations : the first is public, until the second is private
The $\kappa(L^k)$ is the ensemble of parameters starting of f and g, then the public key is f, until the private key must be g .
Given a message $m \in \{0,1\}^n$ and a random oracle r$\leftarrow \{0,1\}^n$, the algorithm of **Encryption** $\varepsilon_{pk}(m,r)$ is calculated as follows :
$S = (m||0^{k_1}) \bigoplus G(r)$ and $t = r \bigoplus H'(S)$, so it is simple to calculate c = f(S,t)
Now to calculate D_{sk} with the algorithm of **Decrypt**, we will do as follows :
(S,t)=g(c) and then we can calculate $r = t \bigoplus H(S)$ and M=S $\bigoplus G(r)$
so if $[M]_{k_1} = 0^{k_1}$ then returned $[M]^n$ if not, returned reject .
The $[M]_{k_1}$ is noted LSB : least significant bit, and thus $[M]^n$ must be reserved to MSB : Most significant bit .
Corresponding transformation :
Suppose we start with the block of plaintext M (the message encrypted) which is a sequence of bytes in length l_n.

- □ **Encoding**. This sequence of bytes M is converted into a sequence of bytes EM with length l_n. The block EM is the encoded message. C is the encoding EME-OAEP which is composed by : EME : Encoding Method for Encryption, and OAEP : Optimal Asymmetric Encryption Padding)).
- □ **Transformation in integer**. Then, EM is transformed into integer m by the procedure OS2IP (Octet String To Primitive Integer).
- □ **Encryption**. The integer m is encrypted in c by the function RSAEP (RSA Encryption Primitive). This transformation is simply the application of the function RSA to the integer m.
- □ **Transformation of the series of bytes** . The integer c is transformed into a sequence of bytes C by the procedure I2OSP (Integer To Octet String Primitive).

When the recipient receives C he applies the following scheme of the decryption

- ✠ **Transformation in integer**. C is transformed into integer c by the procedure OS2IP (Octet String To Integer Primitive) :
- ✠ **Decryption of the integer c is transformed into m RSADP (RSA Decryption Primitive)**. This transformation is the function RSA^{-1} (classic).
- ✠ **The transformation in a serial of bytes :** The integer m is transformed into a sequence of bytes EM by the procedure I2OSP (Integer To String Octet Primitive).
- ✠ **Decoding :** The sequence of bytes EM is converted into a sequence bytes M by the decoding procedure of EME-OAEP.

The procedure OS2IP is very simple. If $X_1, X_2, , X_l$ are the bytes of the string, set x_{l-i} the integer represented by the byte X_i $1 \leq$ i $\leq$ l. The : x = $\sum_{j=0}^{l-1} x_j(256)^j$ is the output of the procedure OS2IP.
The I2OSP procedure is the reverse of the procedure OS2IP corresponds to the decomposition of the entire base 256, which applies to the procedure. For more detail see[12]
For the CPA it is assumed that RSA-OAEP is secure in[13]. We leave the fact to prove that our cryptosystem : version 1 and 2 are secured after integrating a suitable change of variable.

5.2 Security of the third version

We have seen that integrate the RSA-OAEP in Identification Based Encryption, has a lot of problems. The first one is how we can relate the exponent e with an Identity. In[5][6] the authors propose a method which use the random oracle. But their methods has some weakness as we can't be sure if the e_{ID} constructed is prime with $\varphi(n)$ which is a danger because this condition is principal in the RSA. More than that the metho$d$[5][6] suffer from the problem of malleability. By contrast in our proposition (method 3 in section 3.3.1) we will be sure that the e_{ID} is prime with $\varphi(n)$. More as our method is traced in the standard model and as our e_{ID} is prime we would'nt have the problem of malleability.
The second problem of integrating the RSA-OAEP in the IBE is the share of n by a lot of users (in an organization) which may cause the problem of factorizing n. That's why the authors in[5][6] propose the method of SEM. In this method the users has a half unknown key keep with SEM, this block the attack of n. But, this can be realized if SEM is trust which make (perhaps) their method not sure. In addition to these weakness, this method necessitate a lot of authentic, user will be trust (there may exist multi-user which use the help of the SEM too attack the challenge),it is heavy, it doesn't conserve the RSA classic.
In our method of section 4.2 we haven't any of these weakness since : we conserve the RSA classic, we aren't linked to the SEM, all our users are independent : they depend only to the PKG.
The third version using our method to generate e for an ID is secure in the sense of Semantic-CCA2 (noting that Semantic-ATK is equivalent to IND-ATK, according to[21]), because :

1. The RSA-OAEP is secure by[15]
2. The method to generate e for an ID resist to malleability
3. The e_{ID} constructed is standard (we usen't any random)
4. All user are independent

Using this and the following lemma. The third version is Semantic-CCA2 secure
Lemma 1 : IBE-RSA/OAEP system with n users is semantically secure and

$$Succ_n{}^{IBE-RSA}(t_n, q_d, q_e) = Succ_n{}^{RSA}(t_n, q_d, q_e) \leq q_e n Succ_1{}^{IBE-RSA}(t_1, q_d) = q_e n Succ_1{}^{RSA}(t_1, q_d)$$
$$\text{where } t_{_1} = t_n + O(log(q_e n))$$

We need this lemma because we have one n, is share to a lot of identity
Noting that in this lemma we aren't need to the trust user (contrary to[6]). Because we haven't a SEM and our user are independent (we have study this in the section 4.2.1 :i successive and not successive)

5.3 Security of the second version

The security of the leaving version (1 and 2) is related to the security of RSA-OAEP. It is proven that the RSA-OAEP is CCA1 secure and to show CCA2. Shoup[14] noted that it isn't secure but in 2004, Eiichiro Fujisaki, Tatsuaki Okamoto, David Piontcheval, Jacque Stern have founded the opposite result[15], they prove that OAEP-RSA is CCA2 secure. We don't enter into the detail, we assume that the RSA-OAEP is CCA2 secure, but what we can say about our cryptosystem : $\prec (M_{var} + S)^e, S^e \succ$? (version 1 and 2).
Assuming we use the version 2 (after a certain moment the two entities communicated independently from the PKG)
So the two entities send the message in the form $\prec (M + S)^{e_{ID}} mod S', S^{e_{ID}} mod S' \succ$ since we added to our message every time a constant number S (it should be fixed to authenticate the transmitter). So for a sufficient number of result our cryptosystem can't be neither CPA nor CCA1 and CCA2 secure. Then to remedy this problem we propose the following method of mask :
$\prec (M + S + h')^{e_{ID}}, (S + h'')^{e_{ID}}, (h' + h'')^{e_{ID}} \succ$ the functions h ', h" are the proposed masks (variable) As M + S + h ' is equal to cte_i (cte_i is variable for each communication i) and S+h"= cte_i also (the h" is variable)
The decrypt of this message is as follow :
$\prec ((M + S + h')^{e_{ID}})^{d_{ID}}, ((S + h'')^{e_{ID}})^{d_{ID}}, ((h' + h'')^{e_{ID}})^{d_{ID}} \succ = \prec M + S + h', S + h'', h' + h'' \succ$
Then S + h"-(h '+ h") = S-h' and we subtract this by 2S (firstly we must obtain S in the first message before the independence ie t $< t_1$) we so obtain : -S-h'
In the end we calculate M+S+h'-S-h'=M.
After we make this mask, our cryptosystem with OAEP is provided in the form :

$\prec (((M+S+h')^{e_{ID}}))_{OAEP}, (((S+h'')^{e_{ID}}))_{OAEP}, (((h'+h'')^{e_{ID}})^{d_{ID}})_{OAEP} \succ$
For example $(((M+S+h')^{e_{ID}}))_{OAEP}$ means that we applied the method of OAEP to :
$((M+S+h')^{e_{ID}})$. The decryption is after the method of OAEP.
To study the security of this cryptosystem we study firstly the security of
$\prec (M+S+h')^{e_{ID}}, (S+h'')^{e_{ID}}, (h'+h'')^{e_{ID}} \succ$ and after we integrate the security of RSA-OAEP.
Now we have used version 2, so we can begin by the initials messages
$\prec ((m_0+S)^{e_{ID}})_{OAEP}$,$(S^{e_{ID}})_{OAEP} \succ$ which must be circulated for t $< t_1$ (depend on PKG). We are interested only to the $\prec ((m_0+S)^{e_{ID}}, S^{e_{ID}} \succ$ and after we may occurs the RSA-OAEP.
We have so the following system :

$$\begin{cases} \prec (m_0+S)^{e_{ID}} \text{ mod (n) }, S^{e_{ID}} \text{ mod (n)} \succ \\ \prec (M+S+h')^{e_{ID}} \text{ mod (S')} = cte_2, (S+h'')^{e_{ID}} \text{ mod (S')} = cte_2, (h'+h'')^{e_{ID}} \text{ mod(S')} \succ \end{cases}$$

For the PKG it's a system of 4 equations and 4 unknown. The PKG can't decrypt or access to : $\prec (M+S+h')^{e_{ID}}$ mod (S') = cte_2, $(S+h'')^{e_{ID}}$ mod (S')= cte_2, $(h'+h'')^{e_{ID}}$ mod(S') $\succ$. Mathematically it is soluble
For the public it is a system of 4 equations and 6 unknown. Mathematically it is insoluble.
Can we resolve it or extract it some formations if we applies the method of simulation : CPA and CCA (giving the power to the opponents) ?
Firstly we have : CPA $\Rightarrow$ CCA1 $\Rightarrow$ CCA2.
The term $\Rightarrow$ means that this problem is harder than the others, and here the CCA2 is the strongest. So we prove only the rigidity of the system above against CCA2.
As we are interested to integrate the RSA in the IBE our study should be therefore against IN-ID-CCA2[2]. Remembered this technique :
We say that our system is rigid against IN-ID-CCA2 if our algorithm of simulation (instead of the opponent) has no advantage in communicating with the challenger (the verifier or tester). The test is as follows :

Setup :

The challenger chooses the systems of parameters and a master key, he gives the system to the adversary and he leaves the master key to himself.

Phase 1 :

The opponent issued the choice (queries) q_1 ....q_m as follow :

1. **Extraction queries** $\prec ID_i \succ$: The challenger executes his algorithm **Extract** to extract the corresponding keys $\prec d_i \succ$ for every $\prec ID_i \succ$ issued by the opponent and he give it to him
2. **Decrypt** $\prec ID_i, C_i \succ$ **:** More of 1 the challenger executes his algorithm **Extract** to given the corresponding keys $\prec d_i \succ$ for every $\prec ID_i \succ$ he executes also the algorithm **Decrypt** to decrypt any C_i by his corresponding key and he sending the decrypt (clair text) to the opponent

These keys must be chosen adaptively (ie one depend to the other).

Challenge :

The enemy (adversary) choose an identity ID and two messages of the same size m_{t_1} and m_{t_2} as test, with condition that they did not appear in either phase 1 nor 2.
The challenger choose b $\in$ {0,1} he send C=Encrypt(params, ID, m_{t_b}) to the adversary.

Phase 2 :

The adversary issued more q_{m+1} ....q_n .The q_i are :

1. **Extraction queries** $\prec ID_i \succ$ with $ID_i \neq$ ID it is the same as the phase 1.
2. **Decrypt** $\prec ID_i, C_i \succ$ **:** Also same as Phase 1 but on condition that $\prec ID_i, C_i \succ \neq \prec ID, C \succ$.

Those choices must be adaptive.

Guess :

Finally the adversary has responding by $m_{t_{b'}}$ and he win the gain if b' = b.

We said that this test has an advantage adv_t negligible if $adv_t = \mid Pr[b=b'] - \frac{1}{2} \mid < \varepsilon, \forall\, \varepsilon > 0$.
The IND-ID-CCA1 works in that manner, but in the simulation study we don't use the Phase 2.

The security of this version is linked to :

1. Security of the IB^{SEM}-RSA-OAEP
2. Security with multiuser
3. Honest of the user and SEM
4. Security against key Escrow

Staring from this latter

5.3.1 Security against key Escrow

Security Analysis

Assuming that we have two messages of test (m_{t_1} , m_{t_2}) with the same size (we may use different sizes since we calculate according to a mod). We give them to the person who verifies the algorithm of the simulation (which is in place of the enemy). Differently from what we previously seen in principle, in the sequel we give the adversary the power and the possibility to use the **challenge (m_{t_1},m_{t_2})** in his essay. We assume that the challenger chooses for example m_{t_1} he encrypted it and send it to the virtual adversary in the form : $\prec (m_{t_1} + S + h'_{t_1})^{e_{ID}}$ mod (S') = cte_2, $(S + h''_{t_1})^{e_{ID}}$ mod (S')= cte_2, $(h'_{t_1} + h''_{t_1})^{e_{ID}}$ mod(S') $\succ$. We want to know the response of our adversary m_{t_1} or m_{t_2} ?

The virtual adversary is the public

Before giving the message of the test to the opponent he begins his activity but since our S' is obtained from the method of the SEM, so this later can't be replaced. Our opponent may not benefit from the first Extraction i.e give the ID and receive d_{ID} because this be can easily extracted $\varphi(s')$. So we can concentrate only on the second Extraction and we propose to utilize the challenge. We may proceed with the method CPA-CCA2 (in the CCA2 we add in phase 1 and 2 that our adversary may benefit from the **Encrypt** message i.e that the opponent issue the queries $\prec ID_i$,$M_i \succ$ of his chosen to the challenger, this latter encrypt and send it to this opponent). Therefore our adversary has the advantage only to test the expression of $(h' + h'')^{e_{ID}}$ mod(S') which varies. But can we extract something ? the adversary can't draw anything, because we have the sum so we can't separate the associate change to h' of that associate to h" and more the module S' is unknown. Even with this, we can't try and estimate the power of our opponent. So following the principle we give him the challenge (previous message : the encrypt of m_{t_1}), but the opponent who has permission to have more advantage after the challenge and before his response rates to change for example if we have a malignant adversary

$\begin{cases} \checkmark \text{ one bit for example in } m_{t_1} \text{ and two bit for example in } m_{t_2} ; \\ \checkmark \text{ all the bits except one for example in } m_{t_1} \text{ and all the bits except two for example in } m_{t_2}. \end{cases}$

To visualize this we can draw the following diagram :

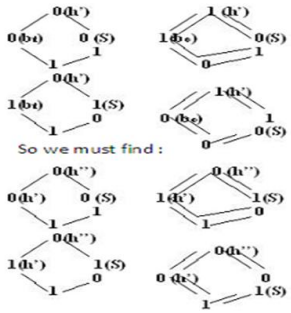

FIG. 7 – Comparison between the originals bits with those that we varied

In this diagrams the single traits are for bits which we change (b_t) and those of the double lines are for the original bits (b_0).
If we choose $b_0 = 1$ ie that the original bit we choose is 1 and if for h' we choose the bit 0 so $1(b_0)$+0(h') = 1. But if we change b_0 by b_t we have : $0(b_t)$+1(h') = 1 (we must conserve the same cte=1 since we want to

test only h'+h"). After an addition with h"=0 we have 0(h')+0(h") = 0 for the original and 1(h')+0(h") = 1 for that we varied.
So basing on comparison, the results reversed (the $(h' + h'')_b$ for the originals are opposed to the test). So if we install the change in either the LSB or MSB or $\frac{LSB+MSB}{i}$ there will be a total change from the original. Because for LSB the change influence on all the follow bits and it is the same for $\frac{LSB+MSB}{i}$ (i < to concerned mod). But if we install the change in MSB it must give us either a message of size smaller than the original or greater than the original (because we calculate according to mod so the difference size can't serve us). The results are so complicated if we expose with e_{ID}, and after the calculation according to the module, therefore the change must be great. So even if we give all that power to our opponent (use : m_{t_1}, m_{t_2}), we will have the total change. And thus we conclude that in addition of S' which remains unknown to our public adversary the $(h' + h'')^{e_{ID}}$ mod(S') is far to be attacked and the opponent can't distinguish whether the encryption is for m_{t_1} or m_{t_2} .

The virtual adversary is the PKG

In this section our algorithm of the simulation is intended to investigate the ability of the PKG. We have noted previously that for
$\begin{cases} \textbf{(1)} \quad \prec (m_0 + S)^{e_{ID}} \text{ mod (n) }, S^{e_{ID}} \text{ mod (n)} \succ \\ \textbf{(2)} \quad \prec (M + S + h')^{e_{ID}} \text{ mod (S')}=cte_2, (S + h'')^{e_{ID}} \text{ mod (S')}=cte_2, (h' + h'')^{e_{ID}} \text{ mod(S')} \succ \end{cases}$
The PKG can solve this system because we have 4 equations and 4 unknown, but in reality no, because even if the PKG can access to **(1)** so to (m_0,S) as we have mod (n), but for the **(2)** she can not access to any of their parameters since we have mod (S') and the S' is not factorial (the PKG can't change S' because this change can be easily seen after the fact that he can not decrypt). So if we pursue the argument cited above the PKG is unable to attack the challenge CCA2 because she wouldn't access to $(h' + h'')^{e_{ID}}$ mod(S').

□

As we show the rigidity of this cryptosystem against Key Escrow we move now to prove the security in general (i.e condition 1,2,3).
But to prove this we need the following lemma :
Lemma 2. IB^{SEM}-RSA/OAEP system in a single-user setting is polynomially as secure as standard RSA/OAEP encryption, i.e., $Succ_1 IB^{SEM}(t, q_d) = Succ_1^R(t', q_d)$ where c is constant value, t' = t + c.
This lemma was given in[6] The interest behind this lemma is to show that attacking RSA (for 1 user) using SEM is equivalent to attack it in classic form (without SEM)
Lemma 3 : IB^{SEM}-RSA/OAEP system with n <u>honest</u> users is semantically secure and

$$Succ_n{}^{IB^{SEM}-RSA}(t_n, q_d, q_e) \leq q_e n Succ_1{}^{IB^{SEM}-RSA}(t_1, q_d)$$
$$\text{where } t_1 = t_n + O(log(q_e n))$$

We need this lemma because in IB^{SEM}-RSA we use one modulus for multiple users
Lemma 4. Under the adaptive chosen ciphertext attack, the system view of the outside adversary (V1), is polynomially indistinguishable from the combined system view (V2) of a set of malicious insiders, in the random oracle model.
With V_1 : := Pr {N, $(e_0, ..., e_n), \Gamma_O, ..\Gamma_E, ..\Gamma_D, ..\Gamma_{SEM}\}$
V_2 : := Pr {N, $(e_0, ..., e_n)$, $\{d_{u_i}\}$,$\Gamma_O, ..\Gamma_E, ..\Gamma_D, ..\Gamma_{SEM}, \Gamma_{d_{u_i}}\}$
where $\{d_{u_i}\}$ is the set of user key-shares, $\Gamma_0, \Gamma_E, \Gamma_D$ are three scripts recording all queries/answers to the random oracle, encryption oracles and decryption oracles, respectively, Γ_{SEM} is the script recording all requests/replies between all users and the SEM ; $\Gamma_{d_u,n}$ is the script recording all n users computation on ciphertexts with their own secret key-share d_{u_i}.
It is shown in[19] that if a Turing Machine $\Im^F$ which access to the random oracle F is undifferentiate from the standard Model G, and $\Im^F$ can replace G in any cryptosystem. The resulting cryptosystem will be at least as secure in the F model as in the G model.
In[6] they use the random oracle in the exponent, but in our proposition we use the standard model.
So after what is shown in[19] we can benefit from this lemma
With our proposition of the exponent we can't use lemma 4 and the proposition1 in[6], because our exponent resist to the malleability.
Lemma 5 : Suppose that there exists an IND-ID-CCA adversary A against our scheme that has an advantage

ε and running time t(k). Suppose also that during the attack A makes at most q_d decryption queries and at most q_e queries of encryptions (allowed to be performed by each user). Then there exists an algorithm B to solve the Decision RSA Short Encrypted-Prime Problem (D-RSA- SEP) problem with advantage $\frac{\varepsilon}{2}$ and t_B= t+$q_d(\tau(exp + mod))$
With $\tau(exp + mod)$ is the maximum time to calculate an exponentiation and a modulus.

Proof

Firstly to generate an exponent e for an identity we use our method instead of that of[6], which is linked to the random oracle. In addition to the fact that with our simple variation of[6] we choose our partial private key so :
Proof. Suppose A has advantage ε in attacking the IB^{SEM}-RSA system. We build an algorithm B that uses A to solve the Decision RSA Short Encrypted-Prime Problem (D-RSA- SEP). Algorithm B's goal is to output 1 if he is able to decide whether or not z_1 is of the form k^{e_1} mod n_1, for some k $\in$ Prime $[2^l, 2^{l+1}]$). and 0 otherwise (response with an arbitrary values).

Setup :

The challenger choose an n_1 (as a modulus) an identity ID_1 for which we can calculate e_1 and a l fixe, he publish then $< n_1$,ID_1, l >

Phase 1 :

A issues up to q_d decryption queries. Algorithm B responds to the query as follows :
Firstly for each $ID_i \neq ID_1$ which is challenged the adversary B follow the the method (3.3.1-third method) to calculate e_{ID_i}
For **Extraction queries of the private key** :
B choose an $d_{partial_{ID_i}}$ he give it to the challenger
With this method B construct a list of tuple e_{ID_i} which he give its $d_{partial_{ID_i}}$ to the chenllenger
For the **Extraction decrypt :** For each ID asked B examine its list
If ID is on the list B demand $c^{d_{SEM_{ID}}}$ from the challenger (note that in this case SEM and challenger are the same)and he can then decrypt the message.
If not B choose again $d_{partial_{ID}}$ he give it to the challenger to receive $c^{d_{SEM_{ID}}}$. In the end he give the plaintex to A

Before constructing the challenge we put that our system can be written as :
$((mh')^e modn, (sh'')^e modn, (h'h'')^e modn)$ instead of $((m + h')^e modn, (s + h'')^e modn, (h' + h'')^e modn)$. The only difference with them is that the first has a bit long compared to the second

Challenge :

The adversary A decides to finish Phase 1, he outputs two equal length plaintexts m_0, m_1 (we can even use different modulus as exponent and we calculate according to a modulus)
B responds with the ciphertext $C_b = (m_b)^{e_1}.z_1 modn_1, sh_b'')^e modn_1, (k_1 h_b'')^e modn_1)$
Note that B can pick any arbitrary $(sh_b'')^e modn_1$ which he has receive from the challenger from phase 1
So if $z_1 = k^{e_1}$ mod n_1 C_b is a ciphertext valid for M_b

Phase 2 :

the same as phase 1 except that B is not allowed to send the decryption of ID_{ch}

Guess :

Finally, A outputs a guess b' $\in \{0, 1\}$ when B output b. If b = b' then B outputs 1 meaning that $z_1 = k^{e_1}$ mod n_1. Otherwise, it outputs 0.

1.If z_1 is not in the form $k^{e_1} modn_1$, i.e., if z_1 is a random in $Z^*_{n_1}$. Then, the challenge ciphertext is not clair, and the simulation will failed. So the advantage of the adversary is necessarily 0, because the ciphertext will be independent to m_b. Therefore, the probability of the failed answer will be $\frac{1}{2}$.
2. If $z_1 = k_1^{e_1}$ mod $n_1$for some l_{k1} -bit prime. So the view of the adversary is normal, and it should have ε as advantage.
In conclusion, we have an advantage $\frac{\varepsilon}{2}$ over the A-D-RSA-SEP problem.

This lemma is proved in the selective-ID which is invented by[20] (it is a weaker security compared to the full one)
Theorem : Our scheme is secure against IND-ID-CCA adversaries. Suppose that there exists an IND-ID-CCA adversary A against our scheme that has an advantage Adv and running time t(k). Then :
Adv $\geq \frac{\varepsilon}{2} {Succ_n}^{IB^{SEM}-RSA}(t_n, q_d, q_e)$
And
t(k)= t_n+$q_d(\tau(exp + mod))$=$t_1 - O(log(q_e n))$+$q_d(\tau(exp + mod))$

6 Comparing our authentication with the Signature of Shamir

The inventor of IBE Shamir didn't succeed to integrate the RSA in the IBE, but he uses it in the scheme of signature[1]. Since this last is designed to authenticate users. What we do in this article is the opposite, we integrated RSA in IBE. But to authenticate entities, we are only based on the integrity of data (version 1 and 2). Can we ensure that with this method, we can get the same level of security as that of Shamir signature ?

5.1 Description of the signature of Shamir

Secret Key : g (secret) such that i = g^e (mod n), i is the identity
S : s=g.$r^{f(t,m)}$, r is chosen arbitrarily by the user, until f is chosen by the PKG
t : t=r^e
We send (s,t), and we verify : $s^e = i.t^{f(t,m)}$mod n

5.2 Comparison

Parameters of Shamir signature	Parameters of our authentication
$\prec s = g.r^{f(t,m)}, t = r^e \succ$	$\prec (m + r)^e, r^e \succ$

For our authentication we choose r instead of S_{Alice} , S_{Bob} to simplified the comparison

For Signer
– Second term : For both they have the same second term r^e
– First term : For **Shamir** it contain the secret parameter g and the expression of s is linked to r, m and the public parameters f, t. For **Our**, the expression : $(m + r)^e$ is also linked to r and m (the message) in addition of the public parameter e.
So our authentication has approximatively the same parameters as the signature of Shamir (since this later is large in comparison with Our)

For Verification
© We note that the verification of Shamir is related to i = g^e and t = r^e.
© As long as our authentication is based only on the integrity of data and identification.

Security
® The security of Shamir's signature is linked to the secrets key g and r as i = g^e and t = r^e. So, the security is bound to calculate the e^{ime} root of g^e and r^e.
® Our authentication is linked to the secret parameter r (identification) which we can not extract it unless we calculates also the e^{ime} root of r^e in addition it is also based on the verification of m+r-r=m which necessitate the calculation of the e^{ime} root of $(m + r)^e$.

5.3 Recap

Our authentication has the same level as that of Shamir, because its security is linked like Shamir (who is related to e^{ime} root for two parameters). Our authentication is also linked to the e^{ime} root for 2 parameters,

and we add one identification r, and one integrity of data
m + r - r = coherent message. Since we know very well that those latter are the principals conditions in cryptography in general.

So in both version 1 and 2 we have presented a cryptosystem in which we crypt and sign in the same time. More than that our signature is strong as it has the same level of Shamir signature

7 Conclusion

In **PKI** the trusted authority CA **manages** all extraction of the public parameters (by certificate) and Private parameters (secure channel). But in the IBE there is a public parameter e_{ID} which we do not require the permission of the authority to extract it. Because of this we have the problem if we want to integrate the RSA in the IBE, since we can have the problem to extract the convenient modulus. But according to this article it is not an infeasible problem that we have imagined for about 30 years. Since as we have seen with the technique of (Boneh, Ding, Tsudik), we can make without problem our goal. But the problem with this method is that we are necessarily linked to PKG which knows all our secrets and we are also linked to the help of what we so called SEM for each message. In this article we give the methods to solve this problem and in which we can sign and crypt at the same time. More than that we have presented a new method that is independent totally from the method of SEM. Can we have other methods to integrate the most famous cryptosystem in the world in the recent technique IBE ?

Acknowledge :

We would like to thank gratefully Mr.Abderahmanne Nitaj for the useful feedback and discussion about this Work. And we would like to thank the head of our laboratory Mr Aboutajdinne Driss.

Références

[1] A. Shamir. "Identity Based cryptosystems and signature schemes", in Advances in cryptology-crypto 84 Lecture Note in computer Science, vol. 196, Springer-Verlag, pp.47-53, 1984.

[2] D. Boneh and M. Franklin. "Identification based Encryption from Weil Pairing", Appears in SIAM J. of Computing, Vol. 32, No. 3, pp.586-615, 2003.

[3] M. Bellare and P. Rogaway. Random oracles are practical : a paradigm for designing efficient protocols. In Proceedings of the First Annual Conference on Computer and Communications Security, ACM, 1993.

[4] D. Boneh, X. Ding, G. Tsudik, C. M. Wong. "A method for fast revocation of public key certificates and security capabilities". In 10th USENIX Security Symposium, Washington, D. C., Aug. 2001. USENIX.

[5] D. Boneh, X. Ding, and G. Tsudik. "Identity based encryption using mediated RSA^*". In 3rd Workshop on Information Security Application, Jeju Island, Korea, Aug. 2002. KIISC.

[6] X. Ding and G. Tsudik, "Simple Identity-Based Cryptography with Mediated RSA^*".In Proceedings of CT-RSA'03, volume 2612. Springer-Verlag, 2003.

[7] B. Chevallier-Mames [2006] "Cryptographie à clé publique : Constructions et preuves de sécurité", These. Available on : http ://bcm.crypto.free.fr/pdf/PhD.pdf

[8] M. Jason Hinek. "On the Security of Multi-prime RSA", Cheriton June 13, 2006. Available on : http :// www.cacr.math.uwaterloo.ca / techreports / 2006 / cacr2006-16.pdf

[9] D. Boneh. "Twenty years of attacks on the RSA cryptosystem". Notices of the American Mathematical Society (AMS), 46(2) :203-213, 1999.

[10] M. Kantarcioglu. "Semantic Security of RSA", 4 / 1 / 2008. Available on : http :// www. utdallas.edu/muratk / courses /crypto09s_files/semantic-rsa.pdf

[11] M. Bellare and P. Rogaway. "Optimal asymmetric encryption". In : EURO- CRYPT'94, LNCS, vol. 839, pp. 92-111. Springer, Heidelberg (1994).

[12] RSA Laboratories. "RSAES-OAEP Encryption Scheme", RSA Security Inc. 20 Crosby Drive Bedford, MA 01730 USA.

[13] B. Chevallier-Mames1, M. Joye2. "Chosen-Ciphertext Secure RSA-type Cryptosystems", Published in J. Pieprzyk and F. Zhang, Eds, Provable Security (ProvSec 2009), vol 5848 of Lecture notes in Computer Science, pp. 32-46, Springer, 2009.

[14] V. Shoup. "OAEP Reconsidered".In Crypto '2001, LNCS 2139, pages 239-259. Springer-Verlag, Berlin, 2001.

[15] E. Fijusiki, T. Okamoto, D. piontcheval, J. stern. "RSA-OAEP is Secure under the RSA Assymption" Journal of Cryptology, Volume 17, Number 2, Pages 81-104, Springer-Verlag, 2004.

[16] A. Granville. "Nombres premiers et chaos quantique", Juin 2002. Available on : http : // smf4.emath.fr/Publications/Gazette/2003/97/sm_gazette_97_29-44.pdf

[17] S. Galbraith. "Mathematics of Public Key Cryptography " book.

[18] "Construction de nombres premiers". Fiche of Acrypta by Ainigmatias Cruptos, Distributed by the Association **ACrypTA**. Available on : http :// www.acrypta.com/telechargements/fichecrypto_111.pdf

[19] U. Maurer, R. Renner, and C. Holenstein, Indiferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology, Theory of Cryptography - TCC 2004, Lecture Notes in Computer Science, Springer-Verlag, vol. 2951, pp. 21-39, Feb 2004.

[20] R. Canetti, Sh. Halevi, and J. Katz. A forward-secure public-key encryption scheme. In Eli Biham, editor, Advances in Cryptology - EUROCRYPT 2003, volume 2656 of Lecture Notes in Computer Science, pages 255-271. Springer-Verlag, 2003.

[21] Pointcheval, and Ph Rogaway. Relations among notions of security for public-key encryption schemes, volume 1462 Lecture Notes in Computer Science, pages 26-45. Springer-Verlag, 1998.

Appendix :

A. How many primes are less than one million ? less than one billion ? less than any given numbers ?

x	number of the number primes up to x
10^8	5761455
10^9	50847534
10^{10}	455052511
10^{11}	4118054813
10^{12}	37607912018
10^{13}	346065536839
10^{14}	3204941750802
10^{15}	29844570422669
10^{16}	279238341033925
10^{17}	2623557157654233
10^{18}	24739954287740860
10^{19}	234057667276344607
10^{20}	2220819602560918840

This table is extracted from[16].

A.1 Our Estimation about the number of primes less than a given number

From this table we see that we have a sequel in the number of the numbers primes, since we started with seven decimal symbol for 10^8 (8-1) and we arrived at the number of nineteen decimal symbol for 10^{20} (20-1). So if we generalize we find that for 10^{2048} we can find that the number of the number prime is a number of at least (2048-1=2047 decimal symbol) and for 10^{8192} we can find a number of number prime nearest to (8192-1 = 8191 decimal symbol).
We want now to estimate the number of the number prime up to 2^{2048}. As $10=2^3 + 2 < 2^4 \Rightarrow$

$10^{512} < 2^{4\times512} = 2^{2048}$. So we are sure that the number of the number prime up to 2^{2048} is a number of at least 511 decimal symbols
Imagining now that for a number up to 2048-bit, the number of the number prime is a number of at least 511 decimal symbols ! So we can vastly choose our numbers as we have such size.

A.2 Estimation sure for the density of the prime number

Definition : Let X ∈ N, then $\pi(X)$ is defined to be the number of primes 1 < p < X (i.e up to X).
An important conjecture in XIX^{th}, proposed by Adrien-Marie Legendre and Carl Friedrich Gauss, that $\pi(X)$ is asymptotically equal to $\frac{X}{ln(X)}$. So the portion $\frac{\pi(X)}{X}$ tend to 0 when X tend to the infinity with acceleration of $\frac{1}{ln(X)}$. In other words, primes are rather common among the integers.
This information's are extract from[17].

B. Method to generate and to test the Prime number : A survey

The best method we can surveying is that of[18] (for more detail the reader can follow the site of Acrypta :www.acrypta.com) because it is the full one
Determine if a number is prime is called the "Prime" problem, noted $\mathcal{P}$ and according to (Manindra Agarwal, Nitin Saxena, Neeraj Kayal, PRIMES is in P, 2002) this problem is polynomial, but for large size the time of its execution is not desirable, so we use the test of non-probabilistic primarily and if necessary we launch a deterministic algorithm to confirm that the prime candidate found by the probabilistic algorithm is surely prime.

B.1 Test of non-primarily of Miller-Rabin

Theorem of Fermat : If p is prime, for all element a in $(Z/pZ)^\star$ we have $a^{p-1} = 1$ (in $(Z/pZ)^\star$). The converse is false. However we can have this result for the not prime number p (for example the number of Carmichael).
An improved method leads to the test of Miller-Rabin. This test is a test of non primarily, so if this test answered that a number is not prime then it is certain that this number is not prime. It may also response that this number is probably prime. In this case, the probability to not been detected prime is not fine : we can imposing that such probability is less than 2^{-100}. Such algorithm is called **Monte-Carlo**
The next test is a consequence of two theorem principal : the theorem of **Miller** and that of **Rabin**. For more details see the article we announce in the highest

B.1.1 Witness of Miller :

Let's an odd integer > 1. posing n-1=$2^s t$, with t is odd. If it exists a (1<a<n) such that :
$a^t \neq 1$ (n) and $a^{2^i t} \neq$ -1 ∀ i=0,..,s-1,
So n is not prime
Definition : An element a that verified the conditions of the theorem of Miller which therefore provides evidence of non-primarily of n is called **Witness of Miller** relative to n.

B.1.2 Theorem of Rabin :

Lets n be an odd integer composite and > 9. Posing n-1=$2^s t$ with t is an odd prime, the integers a which satisfy the condition :
$a^t \equiv 1$(n) or verify one of the condition $a^{2^i t} \equiv 1$(n) ($0 \leq i \leq$ s-1) are numbered of at most $\frac{\phi(n)}{4}$

B.1.3 Test of Miller Rabin :

We choose at random (a < n) and we calculate a^t mod n.
If we find 1 then a is not a witness of Miller for n, by contrast we calculate the numbers :
a^{2it} mod n, if for some i we find -1 then a is not a witness of Miller for n. We do this test with k random

values of a, if none of the values that we drawn at random is the witness of Miller, the number n is probably prime. More precisely if n is composite, the probability of being prime is
$< \frac{1}{4^k}$. We may take for example k = 50.

B.2 Construction of the prime number with a given size :

To construct the prime number p with given size we select at random the number of odd size and we put it in the variable X. We test whether the content X is a prime number if it isn't, we added to it 2. Indeed it would be better to remove an odd number of given size at random.