VERSITA

TATRA MOUNTAINS
Mathematical Publications

DOI: 10.2478/v10127-012-0043-7
Tatra Mt. Math. Publ. **53** (2012), 137–146

CCA-SECURE KEY ENCAPSULATION MECHANISM BASED ON FACTORING ASSUMPTION

Gyöngyvér Márton

ABSTRACT. In this article a key encapsulation mechanism is presented which is based on squaring function, where the input element is from QR_N^+, where QR_N^+ denotes the signed quadratic residue group, and N is a Blum integer. The article presents the soundness, the efficiency and the proof of CCA security of the proposed mechanism.

1. Introduction

The security of encryption systems depends on the resistance of these systems against well defined attacks. Between these attacks the most important is the chosen ciphertext attack. According to the results of current researches the encryption systems that are secure against chosen ciphertext attacks or can be converted in such systems, became very important. The newer literature [4] introduces the key encapsulation and data encapsulation concepts which could potentially replace the classical public-key encryption systems.

R a b i n in [6] introduced the squaring function, and proved that inverting this function is as hard as factoring integers. On the other hand, it is well known that the basic Rabin cryptosystem based on squaring function is completely insecure against a chosen ciphertext attack. The proposed key encapsulation mechanism is based on squaring function too, with the differences that the input element is from QR_N^+, where QR_N^+ denotes the signed quadratic residue group, and N is a Blum integer. This group was used by H o f h e i n z et al. [5], to construct a public-key encryption system which is secure against a chosen ciphertext attack. To achieve this property the authors modified the Blum-Goldwasser public-key encryption system, published by B l u m and G o l d w a s s e r in 1985 [3]. This system uses multiple squaring to encapsulate the randomly generated key,

2010 M a t h e m a t i c s S u b j e c t C l a s s i f i c a t i o n: 94A60.
K e y w o r d s: public-key encryption, chosen ciphertext security, factoring assumption.
This research was supported by the Romania KPI-Grant, IPC:1/18/12.

and the additional element added for consistency check is also calculated with high exponent.

The proposed key-encapsulation mechanism in contrast with the mechanism proposed in [5] uses only one squaring on randomly generated key after which it determines its hash value. According to this, the additional element is calculated with one squaring. The security of a system against a chosen ciphertext attack can be proved by using the factoring assumption and the target-collision resistance properties of the underlying hash functions.

2. Basic definitions

2.1. Important attacks

Cryptography recognizes the following two most important attacks:

- Chosen plaintext attack (CPA): A system that is secure against a chosen plaintext attack ensures security against a passive adversary. It is said that such a system is CPA-secure. To achieve CPA-security one can introduce randomization techniques.
- Chosen ciphertext attack (CCA): A system that is secure against a chosen ciphertext attack ensures security against an active adversary. It is said that such a system is CCA-secure. To achieve CCA-security one can check the consistency of the ciphertext.

In 1998 Bleichenbacher [1] successfully attacked RSA encryption and since that the cryptography community accepted that the right security requirement on a public-key encryption system is to achieve CCA-security.

2.2. Public-key encryption system

In 2003 Cramer and Shoup [4] gave a general definition for a CCA-secure hybrid public-key encryption system (HPKE). They introduced the notion of key encapsulation mechanism (KEM) that can be used to encapsulate, to encrypt a randomly generated key using public-key encryption techniques. After this the key can be used in symmetric key encryption system (SKE) which encrypts the actual data. They proved that by combining a CCA-secure KEM with a CCA--secure SKE the result will be a CCA-secure HPKE.

Definition 1. A CCA-secure hybrid public-key encryption system is defined together with 3 algorithms, on (K, M, C):

- The **key generation** algorithm is a randomized algorithm, that generates a uniform and random public-key pk and secret-key sk, when inputted 1^k. It is denoted: $(pk, sk) \xleftarrow{R} Gen_{PKE}(1^k)$, where $(pk, sk) \in K \times K$, and k is

the security parameter of the system, in most cases indicates the length of the key.

- The **encryption** algorithm $Enc_{PKE}(pk, m)$ consists of two algorithms.
  - The **key encapsulation** is a CCA-secure randomized algorithm, when inputted pk, it produces a randomly generated key, and its cipher value. It is denoted by $(key, cipher) \leftarrow Enc_{KEM}(pk)$, where key $\overset{R}{\leftarrow} M$.
  - The **data encapsulation** is a CCA-secure SKE, when key and m are inputted, it computes the $cipher$ value of m: $c \leftarrow Enc_{SKE}(key, m)$.
  - The output of $Enc_{PKE}(pk, m)$ is $(c, cipher)$.
- The **decryption** algorithm $Dec_{PKE}(sk, c, cipher)$ does the followings:
  - If $(c, cipher)$ is not a proper encryption, it outputs REJECT, and stops.
  - If the output of **key decapsulation** algorithm Dec_{KEM} is REJECT, then decryption algorithm outputs REJECT, and stops.
  - Else
    - run key decapsulation algorithm which is a deterministic algorithm, it takes input sk, $cipher$ and computes key

      $$(key) \leftarrow Dec_{KEM}(sk, cipher),$$

    - run the **data decapsulation** algorithm which is a deterministic algorithm, it takes input key and c, $m \leftarrow Dec_{SKE}(key, c)$,
    - the output of $Dec_{PKE}(sk, c, cipher)$ is m.

In practice the key encapsulation mechanisms' security always depends on widely believed, but not proved assumptions. The proposed KEM security is based on the factoring assumption and on the target-collision resistance property of the underlying hash functions.

In order to introduce these notions the definition of negligible function is needed.

2.3. Negligible functions

Definition 2. A function $f(\cdot)$ is negligible if for every polynomial $\epsilon(\cdot)$ there exists an integer n_0 such that for all integers $n > n_0$ it holds that $f(n) < \frac{1}{\epsilon(n)}$.

2.4. Factoring assumption

In cryptography one of the most commonly used assumption is the factoring assumption.

The factoring assumption assumes that the following function is negligible in k for all probabilistic polynomial-time (PPT) F algorithms

$$\mathrm{Adv}_{FAC,F}(k) = \Pr\big[(N, P, Q) \leftarrow Gen(1^k) : F(N) = \{P, Q\}\big],$$

where $Gen(1^k)$ is a PPT algorithm that returns N, P, Q with the following properties:

- $N = P \cdot Q$,
- $P = Q = 3 \pmod 4$, namely P and Q are Blum integers,
- $P = 2p + 1, Q = 2q + 1$, where p, q are odd prime numbers, namely P and Q are safe primes,
- the bit length of p and q is $k/2 - 1$: $|p| = |q| = k/2 - 1$.

2.5. The group of signed quadratic residues

Nowadays more systems use the group of signed quadratic residues, which has some very useful properties.

Let $QR_N^+ = \{|x| : x \in QR_N\}$ denote the set of signed quadratic residues modulo N, where $|x|$ denotes the absolute value of x, and QR_N denote the set of quadratic residues modulo N: $QR_N = \{x \in \mathbb{Z}_N^* : \exists y,\, y^2 = x \pmod N\}$, and N is a composite number generated with $Gen(1^k)$, specified above. This set has some interesting properties, proved already in [5]:

- QR_N^+ with multiplication is a group, and the group order is $\phi(N)/4$,
- membership in QR_N^+ can be decided efficiently,
- computing square roots in QR_N^+ is equivalent to factoring N,
- squaring in QR_N^+ is a permutation over QR_N^+,
- QR_N^+ is cyclic,
- a uniformly and randomly generated g element will be a generator except with the probability

$$(p + q - 1)/(p \cdot q) \leq 2^{-k+2}.$$

2.6. Target-collision resistant property

To prove the CCA-security of the proposed mechanism a target-collision resistant hash function is used.

Let $TR\colon X \to Y$ be a target-collision resistant hash function. The following function is defined, for an algorithm B

$$\mathrm{Adv}_{TCR,B}(TR) = \Pr\big[x \leftarrow X, y \leftarrow B(x) : x \neq y,\, TR(x) = TR(y)\big].$$

For a target-collision resistant hash function it is true that for all PPT adversaries, the function $\mathrm{Adv}_{TCR,B}(TR)$ is negligible in k.

The proposed scheme uses two target-collision resistant hash function, the first is $T\colon QR_N^+ \to \{0,1\}^{l_T}$, and the second is $H\colon QR_N^+ \to \{0,1\}^{l_H}$.

3. The proposed KEM

3.1. The scheme

In order to present the scheme the key generation, the key encapsulation, and the key decapsulation algorithm are presented.

The **key generation** algorithm $Gen_{KEM}(1^k)$ is a randomized algorithm which generates a uniform and random public-key pk and secret-key sk when it is inputted as follows:

- let N be a composite number generated with $Gen(1^k)$, which is described in Section 2.4,
- let $g \xleftarrow{R} QR_N^+$, and $X = (g^{\alpha \cdot 2^{l_T}})^2$, where $\alpha \xleftarrow{R} \{1, \ldots, (N-1)/4\}$,
- the algorithm's output is $pk = (N, g, X)$, and $sk = (\alpha)$.

The **key encapsulation** algorithm Enc_{KEM} operates on input pk, and does the following:

- choose $r \xleftarrow{R} \{1, \ldots (N-1)/4\}$,
- computes $G = g^{r \cdot 2^{l_T}}$, and $key = H(G)$, where $H \colon QR_N^+ \to \{0,1\}^{l_H}$ is a target-collision resistant hash function,
- computes $S = (g^t \cdot X)^r$, where $t = T(G^2)$, and $T \colon QR_N^+ \to \{0,1\}^{l_T}$, is a target-collision resistant hash function,
- the algorithm's output is $(key, cipher)$, where $cipher = (G^2, S)$.

The **key decapsulation** algorithm Dec_{KEM} operates on input α as secret-key and (G^2, S), and does the following:

- if $(G^2, S) \in QR_N^+ \times QR_N^+$ does not hold, the algorithm outputs REJECT, and stops,
- computes $t = T(G^2)$,
- if
$$S^{2^{1+l_T}} = (G^2)^{t+\alpha \cdot 2^{1+l_T}} \tag{1}$$
does not hold, the algorithm outputs REJECT, and stops,
- else
  - because $0 < t < 2^{l_T}$ it follows that $c < l_T$, so using the extended Euclidean algorithm a, b, c can be calculated from the following Diophantine equation
$$2^c = a \cdot t + b \cdot 2^{1+l_T},$$
where 2^c is the greatest common divisor of t, and 2^{1+l_T}.
  - the algorithm's output is (key), where
$$key = H\Big(\big(S^a \cdot (G^2)^{b-a\cdot\alpha}\big)^{2^{l_T-c}}\Big).$$

3.2. The soundness of the scheme

The soundness of the scheme can be verified by elementary calculations.

It is clear that the values of S, and G^2 can be written in the following form

$$S = (g^t \cdot X)^r = g^{r\cdot(t+2\cdot\alpha\cdot 2^{l_T})}, \quad G^2 = g^{2\cdot r\cdot 2^{l_T}}.$$

To verify the correctness of equation (1) one can calculate

$$S^{2^{1+l_T}} = g^{2\cdot 2^{l_T}\cdot r\cdot(t+2\cdot\alpha\cdot 2^{l_T})}, \qquad (G^2)^{t+\alpha\cdot 2^{1+l_T}} = g^{(t+\alpha\cdot 2\cdot 2^{l_T})\cdot 2\cdot r\cdot 2^{l_T}}$$

To verify the correctness of key's computation one can calculate

$$\begin{aligned} S^a \cdot (G^2)^{b-a\cdot\alpha} &= g^{a\cdot r\cdot t+a\cdot r\cdot 2\cdot\alpha\cdot 2^{l_T}} \cdot g^{b\cdot 2\cdot r\cdot 2^{l_T}-a\cdot\alpha\cdot 2\cdot r\cdot 2^{l_T}} \\ &= g^{r\cdot(a\cdot t+b\cdot 2\cdot 2^{l_T})} = g^{r\cdot 2^c}, \\ \left(S^a \cdot (G^2)^{b-a\cdot\alpha}\right)^{2^{l_T-c}} &= g^{r\cdot 2^{l_T}}. \end{aligned}$$

3.3. The security of the scheme

To prove the chosen ciphertext security of the proposed scheme the following game is defined.

The game is defined between a polynomial-time, probabilistic adversary A and its environment, called the challenger with the following steps:

(1) the challenger chooses the functions T, and H,

(2) the challenger runs $Gen_{KEM}(1^k)$ to obtain $\big((N, g, X), \alpha\big)$,

(3) A is given (N, g, X), who may query $H(\cdot)$, $T(\cdot)$, and $Dec_{KEM}(\alpha, \cdot)$,

(4) A queries the challenger, and the challenger computes:
- $(key^*, cipher^*) \leftarrow Enc_{KEM}(pk)$,
- $u \xleftarrow{R} QR_N^+$,
- let (key^*) denote c_0,
- let $\big(H(u)\big)$ denote c_1,
- the challenger responds with the challenge ciphertext $(c_b, cipher^*)$, where $b \xleftarrow{R} \{0, 1\}$,

(5) the adversary A may query $H(\cdot)$, $T(\cdot)$, and $Dec_{KEM}(\alpha, cipher)$, with the only constraint that $cipher \neq cipher^*$,

(6) the adversary A outputs $\hat{b} \in \{0, 1\}$.

Let $Expr(0)$ denote the event that the adversary outputs 1 when $b = 0$, and let $Expr(1)$ denote the event that the adversary outputs 1 when $b = 1$.

Henceforward the following function can be defined as follows

$$\mathrm{Adv}_{KEM,A}(k) = \frac{1}{2}\big|\Pr\big[Expr(0)\big] - \Pr\big[Expr(1)\big]\big|.$$

Definition 3. It is said that the proposed KEM is CCA-secure if for all PPT adversaries A the function $\mathrm{Adv}_{KEM,A}(k)$ is negligible in k.

Now the following theorem can be stated

Theorem 1. *If in the above game T and H are target collision resistant hash functions and the factoring assumption holds, then the function* $\mathrm{Adv}_{KEM,A}(k)$ *is negligible in k, for all PPT adversaries A.*

In order to prove Theorem 1, first Theorem 2 and Theorem 3 will be proved.

Theorem 2. *If D is an algorithm that distinguishes between $\big(N, G^2, H(G)\big)$ and $\big(N, G^2, H(u)\big)$, where G is the value obtained in key encapsulation, and $u \xleftarrow{R} QR_N^+$, and H is target-collision resistant hash function, then we can construct an F algorithm that factors N. It is said that such an algorithm has D-property.*

Formally, this means the following: if the function $\mathrm{Adv}_{DDS,D}(k)$ *is negligible in k, where*

$$\mathrm{Adv}_{DDS,D}(k) = \Big|\Pr\Big[D\big(N, G^2, H(G)\big) = 1\Big] - \Pr\Big[D\big(N, G^2, H(u)\big) = 1\Big]\Big|,$$

then such a D algorithm does not exit.

P r o o f. Let D be an algorithm having D-property. The D algorithm can be used to break the factoring assumption:

- an input to the algorithm D is the tuple (N, G^2, V), where V is either $H(G)$ or $H(u)$, where $u \xleftarrow{R} QR_N^+$,
- D queries hash function H,
- at any point if D queries G to H, then it must be satisfied that D had computed G from G^2, because we assumed that H is a target-collision resistant hash function,
- if this happens with non negligible probability, then it must happen with non negligible property too, that D had computed G from G^2,
- if D can compute G from G^2 with non negligible property, then D can solve the factorization problem with non negligible property, [6],
- but this is a contradiction to the factoring assumption.

Based on the above it can be stated that

$$\mathrm{Adv}_{DDS,D}(k) \leq \mathrm{Adv}_{FAC,F}(k) + \mathrm{Adv}_{TCR,B}(k).$$

So the $\mathrm{Adv}_{DDS,D}(k)$ function is negligible in k. □

Theorem 3. *If A is an adversary that breaks the proposed system CCA-security, then it can construct a D algorithm that has D-property or a B algorithm that breaks the target collision resistant property of T.*

Formally, this means the following

$$\mathrm{Adv}_{KEM,A}(k) \leq \mathrm{Adv}_{DDS,D}(k) + \mathrm{Adv}_{TCR,B}(k) + f(k),$$

where $f(k)$ is a negligible function.

P r o o f. To prove Theorem 3 it is proceeded as in [5].

A D algorithm is constructed such that simulates the input of A. It is known that the input of D is $\big(N, G^2, H(G)\big)$ or $\big(N, G^2, H(u))\big)$. D will challenge A on input (N, g, X), as public-key and on input:

- $(key^*, cipher^*)$, if the input of D is $\big(N, G^2, H(G)\big)$, or
- $\big(H(u), cipher^*\big)$, if the input of D is $\big(N, G^2, H(u)\big)$,

where $g, X, key^*, cipher^*$ are computed by D in the following way:

- $g \xleftarrow{R} QR_N^+$, and $\beta \xleftarrow{R} \big\{1, \ldots (N-1)/4)\big\}$,
- $t^* = T(G^2)$, $X = g^{\beta \cdot 2 \cdot 2^{l_T} - t^*}$,
- $cipher^* = (R^*, S^*)$, where $R^* = G^2$, and $S^* = (G^2)^\beta$,
- $key^* = H(G)$.

Since g is a generator with high probability, then it can be assumed that:

- X can be seen as $(g^{\alpha \cdot 2^{l_T}})^2$, where α is unknown, but can be set as
$$\beta - t^*/(2 \cdot 2^{l_T}),$$
- R^* can be seen as $g^{r^* \cdot 2 \cdot 2^{l_T}}$, where r^* is unknown,
- S^* can be seen as $(g^{t^*} \cdot X)^{r^*}$, because
$$\big(g^{t^*} \cdot X\big)^{r^*} = \big(g^{t^*} \cdot g^{\beta \cdot 2 \cdot 2^{l_T} - t^*}\big)^{r^*},$$
- key^* can be seen as $g^{r^* \cdot 2^{l_T}}$.

When A submits $cipher = (R, S)$ to D to decapsulate the key, D can check if $cipher$ is consistent or not, simply verify the following

$$S^{2 \cdot 2^{l_T}} = R^{t - t^* + \beta \cdot 2 \cdot 2^{l_T}},$$

where t can be calculated because $t = T(R)$. If (R, S) is not consistent, algorithm rejects the input and stops.

If (R, S) is consistent, D will distinguish two cases:

- $t \neq t^*$. In this case D computes a_1, b_1, c_1 from the following Diophantine equation, using the extended Euclidean algorithm:
$$2^{c_1} = a_1 \cdot (t - t^*) + b_1 \cdot 2 \cdot 2^{l_T}.$$

The key can be determined from

$$\left(S^{a_1} \cdot R^{b_1 - a_1 \cdot \beta}\right)^{2^{l_T - c_1}}.$$

- $t = t^*$. This case has two subcases:
  - $R = R^*$. From this follows that $S = S^*$. But it is not allowed to query (R^*, S^*).
  - $R \neq R^*$. In this case D finds a collision, but this can happen only with negligible probability.

As a consequence, pk and (R^*, S^*) are distributed identically in this simulation and the game described in Section 3.3 except with the probability 2^{-k+3}, which has already been proved in [5].

So it is true that:

$$\left|\Pr\left[D\big(N, G^2, H(G)\big) = 1\right] - \Pr\big[Expr(0)\big]\right| \leq \mathrm{Adv}_{TCR,B} + 2^{-k+3},$$
$$\left|\Pr\left[D\big(N, G^2, H(u)\big) = 1\right] - \Pr\big[Expr(1)\big]\right| \leq \mathrm{Adv}_{TCR,B} + 2^{-k+3},$$

where $Expr(0)$ and $Expr(1)$ has the same meaning as in Section 3.3.

It follows that:

$$\begin{aligned}
\mathrm{Adv}_{KEM,A}(k) &= \frac{1}{2}\left|\Pr\big[Expr(0)\big] - \Pr\big[Expr(1)\big]\right| \\
&\leq \frac{1}{2}\Bigg(\mathrm{Adv}_{DDS,D} + \left|\Pr\left[D\big(N, G^2, H(G)\big) = 1\right] - \Pr\big[Expr(0)\big]\right| \\
&\quad + \left|\Pr\big[D(N, G^2, u) = 1\big] - \Pr\big[Expr(1)\big]\right|\Bigg) \\
&\leq \mathrm{Adv}_{DDS,D} + \mathrm{Adv}_{TCR,B} + 2^{-k+3},
\end{aligned}$$

where it is known that:

- $\mathrm{Adv}_{TCR,B}$ is negligible in k (Section 2.6),
- $\mathrm{Adv}_{DDS,D}$ is negligible in k (Section 3.3, Theorem 2).

As a consequence, according to Definition 3 the proposed key encapsulation mechanism is CCA-secure. □

3.4. The efficiency of the scheme

Generating safe prime is quite time consuming, but it can be sped up using the method proposed in [7].

The key encapsulation algorithm uses two modular exponentiations with large exponent, and some multiplications and exponentiation with small exponents.

The key decapsulation algorithm uses one modular exponentiation with large exponent, and some multiplications and exponentiations with small exponents.

This algorithm can be sped up using the Chinese Remainder Theorem if the prime numbers P, and Q are added as inputs of the algorithm.

In consequence the key decapsulation mechanism is twice as efficient as the key encapsulation, which is very attractive, because, in general, in the real application, in many protocols decapsulation is done by a server.

4. Conclusion

The article presented a novel key encapsulation mechanism, that is an extension of the original Rabin cryptosystem, and that chosen ciphertext security proof is based on the proof presented by H o f h e i n z et al. in [5].

REFERENCES

[1] BLEICHENBACHER, D.: *Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1*, in: Advances in Cryptology—CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, CA, USA, 1998 (H. Krawczyk, ed.), Lect. Notes Comput. Sci. Vol. 1462, Springer-Verlag, Berlin, 1998, pp. 1–12.

[2] BLUM, L.—BLUM, M.—SHUB, M.: *Comparison of two pseudo random number generators*, in: Advances in Cryptology—CRYPTO '82, Santa Barbara, California, USA, 1982 (T. Sherman et al., eds.), Plenum Press, New York, 1983, pp. 61–78.

[3] BLUM, M.—GOLDWASSER, S.: *An effcient probabilistic public-key encryption scheme which hides all partial information*, in: Advances in Cryptology—CRYPTO '84 Santa Barbara, California, USA, 1984 (G. R. Blakley et al., eds.), Lect. Notes Comput. Sci. Vol. 196, Springer-Verlag, Berlin, 1985, pp. 289–302.

[4] CRAMER, R.—SHOUP, V.: *Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack*, SIAM J. Comput. **33** (2003), 167–226.

[5] HOFHEINZ, D.—KILTZ, E.—SHOUP, V.: *Practical chosen ciphertext secure encryption from factoring*, J. Cryptology, Online First^{TM}, 19 December, 2011, 1–17.

[6] RABIN, M.: *Digitalized signatures as intractable as factorization*, Technical Report MIT/LCS/TR-212, MIT Laboratory for Computer Science, January 1979.

[7] WIENER, M. J.: *Safe prime generation with a combined sieve*, Crypto Eprint Archive entry 2003:186.

Received August 31, 2012

Department of Mathematics and Informatics
Faculty of Technical and Human Sciences
Sapientia University
sos. Sighisoarei 1C
RO–540-485 Tîrgu-Mureş/Corunca
ROMANIA
E-mail: mgyongyi@ms.sapientia.ro