III. Composite-Residuosity Based Cryptography: An Overview

Pascal Paillier

Cryptography Group, Gemplus

pascal.paillier@gemplus.com

1. INTRODUCTION

There are three main families of public key cryptosystems based on computational number theory. The first family includes RSA and related variants (Rabin-Williams, LUC, Dickson, elliptic curve embodiments of RSA-like KMOV). The trapdoor for these relies on the extraction of roots over finite Abelian groups of some secret order. Root extraction is easy when the group order is known, but believed to be hard without that knowledge. Finding the group order is as hard as factoring a large integer.

The second family is based on Diffie-Hellman-type schemes (ElGamal and variants, Cramer-Shoup) which exploit properties of exponentiation over finite cyclic groups. Here, the trapdoor depends on the knowledge of the discrete logarithm of some public group element and again, computing this secret information from the description of the group alone is believed to be hard.

Finally, the third family is based on high degree residuosity classes (Goldwasser-Micali, Benaloh, Naccache-Stern, Okamoto-Uchiyama and variants). The trapdoor in these schemes combines the extraction of residuosity classes over certain groups with the intractability of computing their order. Because residuosity classes are additive, such cryptosystems look like discrete-log based ones, but the trapdoor is closer in nature to those for factoring-based systems.

We review here one particular cryptosystem belonging to this last family that was proposed by the author of this paper at Eurocrypt '99. The system, based on composite residuosity classes, has recently gained a certain degree of popularity mainly as a building block for cryptographic protocols. We summarize some of these constructions and provide state-of-the-art references to composite-residuosity-based cryptographic tools.

2. DESCRIPTION OF THE SCHEME

We first briefly recall basic facts on composite residues, referring the reader to [8] for more details.

- We set $n = pq$ where p and q are large primes and denote by $\phi(n)$ and $\lambda = \lambda(n)$ the Euler and Carmichael functions of n respectively. Then $\phi(n) = (p\text{-}1)(q\text{-}1)$ and $\lambda = \mathrm{lcm}(p\text{-}1, q\text{-}1)$.
- It is a well-known fact that the group $\mathbb{Z}_{n^2}^*$ has $\phi(n^2) = n\phi(n)$ elements. Furthermore, $w^{\lambda} = 1 \bmod n$ and $w^{n\lambda} = 1 \bmod n^2$ for any element $w \in \mathbb{Z}_{n^2}^*$.
- We say that an integer z is an n-residue modulo n^2 if there exists some $y \in \mathbb{Z}_{n^2}^*$ such that $z = y^n \bmod n^2$. The set of n-residues forms a subgroup of $\mathbb{Z}_{n^2}^*$ of order $\phi(n)$. Each n-residue in $\mathbb{Z}_{n^2}^*$ has exactly n roots of degree n.
- We denote by B the set of elements of order $n\alpha$ for some $\alpha \in [1, \lambda]$.
- Let $g \in \mathbb{Z}_{n^2}^*$ and consider the mapping over $\mathbb{Z}_n \times \mathbb{Z}_{n^2}^*$ defined by:

  $$(x,y) \mapsto g^x y^n \bmod n^2.$$

  Then, when $g \in B$, this map is one-to-one.
- Let $g \in B$. We define the n-residuosity class of an element $w \in \mathbb{Z}_{n^2}^*$ with respect to g as the unique integer $x \in [1, n]$ for which there exists $y \in \mathbb{Z}_{n^2}^*$ s.t. $w = g^x y^n \bmod n^2$.

  Following the notation of Benaloh [2], we denote the class of w by $[w]_g$. Note that $[w]_g = 0$ if and only if w is an n-residue. Additionally,

  $$\forall w_1, w_2, \quad [w_1 \cdot w_2]_g = [w_1]_g + [w_2]_g.$$

  Hence, the class function $w \mapsto [w]_g$ is an additive homomorphism for any $g \in B$.
- Consider the subgroup of $\mathbb{Z}_{n^2}$ defined by $S_n = \{u : u = 1 \bmod n\}$. For $u \in S_n$ define:

  $$L(u) = (u - 1)/n.$$

  Then for $u \in S_n$ we have $L(u^r)/L(u) = r$. Hence, discrete-log is easy in the group S_n.

We are now ready to define the composite residuosity cryptosystem. Let $n = pq$ and $g \in B$. The public key is the pair (n,g) while the factors (p,q) are the private key. The cryptosystem is as follows.

Encryption:

Plaintext: $0 \le m < n$
select a random $0 < r < n$
Ciphertext: $c = g^m r^n \bmod n^2$

Decryption:

Ciphertext: $0 < c < n^2$

Plaintext: $m = \dfrac{L(c^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} \bmod n$

We discuss the security of this system below. This cryptosystem is useful for distributed computations due to its additive homomorphism. That is, for all m_1, m_2 :

$$D(E(m_1)\cdot E(m_2) \bmod n^2) = m_1 + m_2 \bmod n$$

In other words, given the two ciphertexts $E(m_1)$, $E(m_2)$ it is easy to construct the encryption of $m_1 + m_2$.

The additive property is particularly useful when designing threshold crypto-systems and distributed protocols in general. It also allows full self-randomization of encryptions in the sense that any ciphertext can be transformed into another without affecting the plaintext.

3. THE CLASS PROBLEM

We discuss several complexity assumptions needed for the security of the above cryptosystem.

The problem of distinguishing the set of n-residues from non n-residues in Z_{n^2} is denoted by $CR[n]$. This problem has a random-self reduction in Z_{n^2} (reduce $CR[n]$ for a worst case element $x \in Z_{n^2}$ to a random element). The assumption that $CR[n]$ is polynomial-time intractable is referred to as the Decisional Composite Residuosity Assumption (DCRA).

The cryptosystem above is semantically secure (against chosen-plaintext attacks) assuming the DCRA assumption.

The n-Residuosity Class Problem in base g, denoted $Class[n,g]$, is defined as the problem of computing the class function in base g. This is exactly the problem of decrypting a given ciphertext in the cryptosystem above.

As before, one can show that $Class[n,g]$, is random self-reducible over its inputs. Moreover, for any $w \in Z^*_{n^2}$ and $g_1 g_2 \in B$, we have

$$[w]_{g_1} = [w]_{g_2}\ [g_2]_{g_1} \bmod n$$

which implies that $Class[n,g]$ is also random self-reducible over $g \in B$. Hence, $Class[n]$ is a computational number-theoretic problem which only depends on n, very much like factorization for instance. The Class problem is related to other standard hard number theoretic problems. For example, the decryption procedure shows that:

$$\frac{L(w^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} = [w]_g \bmod n.$$

Hence, if we can factor n and obtain λ then we can solve $Class[n]$. Therefore, we write:

$$Class[n] \Leftarrow Fact[n].$$

One can show a slightly stronger statement. $Class[n]$ can be solved just given the ability to compute n'th roots in Z_n. Computing n'th roots in Z_n is called the RSA problem with public exponent $e = n$ and is denoted by $RSA[n,n]$. Therefore, we have:

$$Class[n] \Leftarrow RSA[n,n].$$

In summary, the computational hierarchy behind composite residuosity is

$$CR[n] \Leftarrow Class[n] \Leftarrow RSA[n,n] \Leftarrow Fact[n].$$

We conjecture that $Class[n]$ is polynomial-time intractable; by analogy with the DCRA, this conjecture is called Computational Composite Residuosity Assumption (CCRA for short). We know that if the DCRA is true then the CCRA is true, but the converse implication remains open.

A careful study of $CR[n]$ and $Class[n]$ is essential in future research, because very few things are known about these problems today. We note that the encryption scheme shown above is one-way relative to the CCRA and semantically secure (against chosen-plaintext attacks) relative to the DCRA.

In [4], Catalano, Gennaro and Howgrave-Graham examined the bit security of our scheme. They showed that given a random element $w \in Z_{n^2}^*$, predicting the least significant bit of $[w]_g$ is as hard as computing $[w]_g$ completely. Moreover, they proved that the scheme simultaneously hides $|n|-b$ bits of $[w]_g$ under the assumption that computing classes remains hard over $\{w : [w]_g < 2^b\}$. By encrypting random-padded messages, the authors deduced from their results a way to construct the first encryption scheme hiding $O(|n|)$ plaintext bits. Note that although their modified version of the class problem seems to remain hard in this context, further research on its connections with the original class problem (as well as on possible breakthroughs) is required to validate this approach.

4. CRYPTOGRAPHIC APPLICATIONS

We now give some cryptographic applications of composite residuosity. Without being exhaustive, composite residuosity finds applications in such different fields as encryption, signatures, distributed protocols such as voting schemes and ZK proofs. It is worthwhile noting that among all residuosity-based schemes, taking $g = 1+kn$ for some k leads to higher encryption rates as $g^m = 1+kmn$. Because of random self-reducibility, this choice does not affect the security level.

4.1 A Subgroup Variant

We give here a slightly modified encryption scheme in which the ciphertext space is restricted to the subgroup $\langle g \rangle$. Indeed, assuming that g is of order $n\alpha$, we have for any $w \in \langle g \rangle$,

$$[w]_g = \frac{L(w^\alpha \bmod n^2)}{L(g^\alpha \bmod n^2)} \bmod n.$$

This motivates the following cryptosystem.

Encryption:

Plaintext: $0 \le m < n$
select a random $0 < r < n$
ciphertext: $c = g^{m+rn} \bmod n^2$

Decryption:

Ciphertext: $0 < c < n^2$

Plaintext: $m = \dfrac{L(c^\alpha \bmod n^2)}{L(g^\alpha \bmod n^2)} \bmod n$

This time, the secret key is α instead of λ. The most expensive operation while decrypting is the modular exponentiation $c^a \bmod n^2$, which can be accelerated arbitrarily by an adequate selection of α. In practice, α should be typically set to a 320-bit divisor of λ such that $\alpha = \alpha_p\alpha_q$ where α_p divides p-1 but not q-1 and α_q divides q-1 but not p-1. This can be met using an appropriate key generation algorithm.

In this subgroup variant, one-wayness does not rely on the composite residuosity class problem, because the ciphertext is known to lie in $\langle g \rangle$. The problem consisting in computing residuosity classes in this context is called Partial Discrete Logarithm Problem and is a weaker instance of the class problem. Similarly, we call Decisional Partial Discrete Logarithm Problem the problem of distinguishing n-residues given the public information. The semantic security of the encryption scheme is equivalent to this problem.

4.2 Extended Variant

In [15], Damgård and Jurik introduced a modified cryptosystem in which computations are performed modulo n^{s+1} where $s \geq 1$. Clearly, the original scheme is contained by setting $s = 1$. Damgård and Jurik's extended scheme relies on the observation that for any $g \in Z^*_{n^{s+1}}$ such that n^s divides the order of g modulo n^{s+1}, the function defined over $Z_{n^s} \times Z^*_n$ by

$$(x,y) \mapsto g^x y^{n^s} \bmod n^{s+1}$$

is one-to-one. As a result, n^s-residuosity classes are easily definable in this context and present the same features than in the original system. The particular choice $g = 1 + n$ (it is easily shown that the order of $1 + n$ modulo n^{s+1} is n^s) provides the advantage of reducing the key size without modifying the system's properties (including security). The final observation is that computing m from $w = (1 + n)^m \bmod n^{s+1}$ is easy. Define like in the original setting $L(x) = (x - 1)/n$. Clearly,

$$L((1+n)^m \bmod n^{s+1}) = \left(m + \binom{m}{2} n + \cdots + \binom{m}{s} n^{s+1}\right) \bmod n^s$$

Damgård and Jurik then give an inductive method to compute $m_i = m \bmod n^i$ for successive values of $i \in [1,s]$. A simple alternative to their method is obtained by observing that $(1 + n)^{n^i} = 1 + n^{i+1} \bmod n^{i+2}$ for any i, so we actually have the more direct induction

$$m_{i+1} = m_i + L(w(1+n)^{-m_i} \bmod n^{i+2}),$$

which also allows us to recover $m = m_s$. Damgård and Jurik's cryptosystem is described as follows.

Key Generation: choose an RSA modulus $n = pq$. The public key is n while the secret key is (p,q).

Encryption: given a plaintext $m < n^s$, choose a random $r < n^{s+1}$ and let the ciphertext be

$$c = (1 + n)^m r^{n^s} \bmod n^{s+1}.$$

Decryption: compute d such that $d = 1 \bmod n^s$ and $d = 0 \bmod \lambda$ (this may also be saved as some secret key material). Given the encryption c, compute $c^d = (1+n)^m \bmod n^{s+1}$ and apply the above algorithm to recover m.

The one-wayness of this scheme is based on the assumption that the class function is hard to compute in this context without knowledge of (p,q). Similarly, semantic security is achieved if and only if distinguishing n^s-residues in $Z^*_{n^{s+1}}$ is intractable. These assumptions were called Generalised (Decisional) Composite Residuosity Assumption or G(D)CRA and conjectured true by the authors. It is easily seen that original assumptions imply Damgård and Jurik's generalized assumptions.

4.3 Digital Signatures

Trapdoor permutations are extremely rare objects: we refer the reader to [12] for an exhaustive documentation on these. Here, we show how composite residuosity allows to design a trapdoor permutation. As before, n stands for the product of two large primes and $g \in B$.

Encryption:

plaintext $m < n^2$
split m into $m = m_1 + nm_2$
ciphertext $c = g^{m_1} m_2^n \bmod n^2$

Decryption:

ciphertext $c < n^2$

compute $m_1 = \dfrac{L(c^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} \bmod n$,

compute $c' = cg^{-m_1} \bmod n$,
compute $m_2 = c'^{n^{-1} \bmod \lambda} \bmod n$,
plaintext $m = m_1 + nm_2$.

As easily seen in the decryption procedure, we require the extraction of an n-root modulo n. Because of this additional step, we get that this permutation is one-way if and only if $RSA[n,n]$ is hard. Like with any other trapdoor permutation, digital signatures are obtained by using the cryptosystem backwards : denoting by $\mu: \{0,1\}^* \mapsto \{0,1\}^k$ some padding function with $k = |n^2|$, we obtain a signature scheme as follows. For a given message m, the signer computes the signature (s_1, s_2) where

$$s_1 = \frac{L(\mu(m)^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} \bmod n,$$

$$s_2 = (\mu(m) g^{-s_1})^{n^{-1} \bmod \lambda} \bmod n$$

and the verifier checks that

$$\mu(m) = g^{s_1} s_2^n \bmod n^2.$$

4.4 A Distributed Version

In [15], Damgård and Jurik devised a distributed cryptosystem allowing threshold decryption among a set of servers. Fouque, Poupard and Stern independently proposed a similar technique in [6]. This threshold variant is an adaptation of Shoup's distributed RSA [17] whose main part allows a set of servers to collectively and efficiently raise an input number to a secret exponent modulo an RSA modulus. On input c, each server returns a share of the result, together with a proof of correctness. Given sufficiently many correct shares, these can be efficiently combined to compute $c^d \bmod n$, where d is the secret exponent. Damgård and Jurik transplanted this method in the case of a shared exponentiation modulo n^{s+1}.

Assume that there are l decryption servers, and that a minimum of k of these are needed to make a correct decryption.

Key Generation: pick a pair of primes p and q satisfying $p = 2p'+1$ and $q = 2q'+1$ for primes p' and q'. Set $n = pq$, $m = p'q'$ and decide on some $s > 0$, so that the plaintext space is Z_{n^s}. Then pick an number d to satisfy $d = 1 \bmod n^s$ and $d = 0 \bmod m$. Now choose a polynomial

$$f(X) = \sum_{i=0}^{k-1} a_i X^i \bmod n^s m$$

by picking random coefficients $a_i \in Z_{n^s m}$ for i ranging from 1 to k–1 and $a_0 = d$. The secret share of server i is $s_i = f(i)$ for $i \in [1, l]$ while the public key is n. To verify the actions of the decryption servers, the system requires the following fixed public values: v, generating the cyclic group of squares in $Z^*_{n^{s+1}}$ and for each decryption server a verification key $v_i = v^{\Delta s_i} \bmod n^{s+1}$ where $\Delta = l!$. The entire key setup may be executed by a trusted party, or distributed among servers using suitable multiparty computation techniques.

Encryption: to encrypt a message m, a random $r < n^{s+1}$ is picked and the ciphertext is computed as $c = (1 + n)^m r^{n^s} \bmod n^{s+1}$.

Share Decryption: the server i computes $c_i = c^{2\Delta s_i}$ where c is the ciphertext and provides a zero-knowledge proof that $\log_{c^4}(c_i^2) = \log_v(v_i)$ which shows that he has indeed raised c to his secret exponent s_i.

Share Combining: given a subset S of k (or more) shares with a correct proof, the result is obtained by combining the shares into

$$c' = \prod_{i \in S} c_i^{2\lambda_{0,i}^S} \bmod n^{s+1} \text{ where } \lambda_{0,i}^S = \Delta \prod_{j \in S - i} \frac{-i}{i-j}.$$

We then get

$$c' = c^{4\Delta^2 f(0)} = c^{4\Delta^2 d} = (1 + n)^{4\Delta^2 m} \bmod n^{s+1}.$$

The plaintext m is retrieved by applying the induction formula described in the extended variant and multiplying the result by $(4\Delta^2)^{-1} \bmod n^s$.

The authors then showed that this threshold version is as secure as their extended variant in the random oracle model, provided that some trusted player performs the share combining stage. More recently, Damgård and Koprowski proposed a new threshold RSA technique [16] applicable *mutatis mutandis* to the present setting. Its main advantage over [15] resides in that no trusted dealer is needed.

4.5 Other Applications

Boneh and Franklin [3] introduced a traitor tracing scheme in which black box tracing is achieved using the subgroup variant.

Pointcheval and the author of these lines [10] proposed security-enhanced cryptosystems provably semantically secure against chosen-ciphertext attacks in the random oracle model.

In [14], Poupard and Stern use the subgroup encryption scheme to devise proofs of knowledge for the factorization of a public composite integer. In [13], the same authors further achieve fair encryption of secret keys, a clever and efficient approach to key recovery systems.

Yung and I considered self-escrowed public-key infrastructures [11], in which a joint use of Paillier and ElGamal encryption schemes leads to a simplified implantation of PKI properties.

Cramer, Damgård and Nielsen [5] propose a way of basing multiparty computation protocols on homomorphic threshold crypto-systems instead of using secret sharing schemes. Their general construction is shown to reach a better efficiency in that fewer bits are needed to be transmitted between parties, while security against cheating is preserved for any minority of cheaters.

Galbraith [7] recently showed how to securely design a composite-residuosity-based encryption scheme on non-specific elliptic curves over rings. This implicitly provided an answer to the quest of [9].

More recently, Baudron and Stern [1] exploited our scheme's homomorphic property to design a new auction protocol where bids are submitted non-interactively and bidders are not required to interact with each other.

Even more recently, Cramer and Shoup used our scheme to propose an encryption scheme secure against active adversaries in the standard model [6]. They based their scheme's security on the decisional composite residuosity assumption.

5. IMPLEMENTATION ISSUES

5.1 Efficiency

The reader may find in [8] some tips about practical aspects of computations required by composite residuosity-based cryptosystems, as well as various implementation strategies for increased performance. We just recall here the main tricks: decryption allows Chinese remaindering; preprocessing can be used advantageously in both encryption and decryption; small values for g or setting $g = 1+n$ (which does not affect security at all) would greatly improve encryption rates, provided that $g \in B$ still holds. In the same spirit as with RSA, simple randomization techniques may help protect hardware or software implementations against side-channel attacks.

REFERENCES

1. Olivier Baudron and Jacques Stern. Non-interactive private auctions. In *Financial Crypto'01*, Lecture Notes in Computer Science, pages 300-313. Springer-Verlag, 2001.

2. Josh Cohen Benaloh. Verifiable Secret-Ballot Elections. PhD thesis, Yale University, 1988.

3. Dan Boneh and Matthew Franklin. An efficient public key traitor tracing scheme. In *Crypto '98*, Lecture Notes in Computer Science. Springer-Verlag, 1998.

4. Dario Catalano, Rosario Gennaro, and Nick Howgrave-Graham. The bit security of Paillier's encryption scheme and its applications. In Birgit Pfitzmann, editor, *Eurocrypt* '01, volume 2045 of Lecture Notes in Computer Science, pages 229-243. Springer-Verlag, 2001.

5. Ronald Cramer, Ivan Damgård, and Jesper B. Nielsen. Multiparty computation from threshold homomorphic encryption. In Bart Preneel, editor, *Eurocrypt* '00, volume 1807 of Lecture Notes in Computer Science, pages 280-300. Springer-Verlag, 2000.

6. Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. Available on IACR ePrint Archive

   (http://eprint.iacr.org/2001/085).

7. Pierre-Alain Fouque, Guillaume Poupard, and Jacques Stern. Sharing decryption in the context of voting or lotteries. In *Financial Cryptography 2000*, Lecture Notes in Computer Science. Springer-Verlag, 2000.

8. Steven D. Galbraith. Elliptic curve Paillier schemes. Journal of Cryptology, 2001. to appear.

9. Pascal Paillier. Public-key cryptosystems based on discrete logarithms residues. In *Eurocrypt* '99, volume 1592 of Lecture Notes in Computer Science, pages 223-238. Springer-Verlag, 1999. European patent number 9900341.

10. Pascal Paillier. Trapdooring discrete logarithms on elliptic curves over rings. In T. Okamoto, editor, Asiacrypt'00, volume 1976 of Lecture Notes in Computer Science, pages 573-584. Springer-Verlag, 2000.

11. Pascal Paillier and David Pointcheval. Efficient public-key cryptosystems provably secure against active adversaries. In K. Y. Lam and E. Okamoto, editors, *Asiacrypt* '99, volume 1716 of Lecture Notes in Computer Science, pages 165-179. Springer-Verlag, 1999.

12. Pascal Paillier and Moti Yung. Self-escrowed public-key infrastructures. In JooSeok Song, editor, *ICICS* '99, volume 1787 of Lecture Notes in Computer Science, pages 257-268. Springer-Verlag, 1999.

13. Jacques Patarin and Louis Goubin. Trapdoor one-way permutations and multivariate polynomials. In *ICICS* '97, volume 1334 of Lecture Notes in Computer Science, pages 356-368. Springer-Verlag, 1997.

14. Guillaume Poupard and Jacques Stern. Fair encryption of RSA keys. In Bart Preneel, editor, *Eurocrypt* '00, volume 1807 of Lecture Notes in Computer Science, pages 172-189. Springer-Verlag, 2000.

15. Guillaume Poupard and Jacques Stern. Short proofs of knowledge for factoring. In *PKC* '00, volume 1751 of Lecture Notes in Computer Science, pages 147-166. Springer-Verlag, 2000.

16. Ivan Damgård and Mads Jurik. A generalisation, a simplification and some applications of Paillier's probabilistic public-key system. In Kwangjo Kim, editor, *PKC* '01, volume 1992 of Lecture Notes in Computer Science, pages 119-136. Springer-Verlag, 2001.

17. Ivan Damgård and Maciej Koprowski. Practical threshold RSA signatures without a trusted dealer. In Bart Preneel, editor, *Eurocrypt* '00, volume 1807 of Lecture Notes in Computer Science, pages 152-165. Springer-Verlag, 2000.

18. Victor Shoup. Practical threshold signatures. In Bart Preneel, editor, *Eurocrypt* '00, volume 1807 of Lecture Notes in Computer Science, pages 207-220. Springer-Verlag, 2000.