Osječki matematički list **6**(2006), 37–44

An Introduction to Probabilistic Encryption

Georg J. Fuchsbauer*

Abstract. *An introduction to probabilistic encryption is given, presenting the first probabilistic cryptosystem by Goldwasser and Micali. Furthermore, the required number-theoretic concepts are discussed and the notion of semantic security is presented in an informal way. The article should be comprehensible to students with basic mathematical knowledge.*

Key words: *probabilistic encryption, cryptosystem*

1. Introduction

Historically, encryption schemes were the first central area of interest in cryptography.[1] They deal with providing means to *enable private communication over an insecure channel*. A *sender* wishes to transmit information to a *receiver* over an *insecure channel*, that is a channel which may be tapped by an *adversary*.

Thus, the information to be communicated, which we call the **plaintext**, must be transformed (**encrypted**) to a **ciphertext**, a form not legible by anybody other than the intended receiver. The latter must be given some way to **decrypt** the ciphertext, i.e. retrieve the original message, while this must not be possible for an adversary. This is where keys come into play; the receiver is considered to have a **key** at his disposal, enabling him to recover the actual message, a fact that distinguishes him from any adversary.

An **encryption scheme** consists of three algorithms: The **encryption algorithm** transforms plaintexts into ciphertexts while the **decryption algorithm** converts ciphertexts back into plaintexts. A third algorithm, called the **key generator**, creates pairs of keys: an **encryption key**, input to the encryption algorithm, and a related **decryption key** needed to decrypt. The encryption key relates encryptions to the decryption key. The key generator is considered to be a probabilistic algorithm (see below), which prevents an adversary from simply running the key generator to get the decryption key for an intercepted message. The following concept is crucial to probabilistic cryptography:

Definition 1 [Probabilistic Algorithm]. *A probabilistic algorithm is an algorithm with an additional command* `RANDOM` *that returns "0" or "1", each with*

*e-mail: fuchsbauer@gmx.net

[1] For a profound introduction to cryptography, see [3].

probability 1/2. In the literature, these random choices are often referred to as ***coin flips****.*[2]

Private-Key vs. Public-Key

The first encryption schemes had only one key for encryption and decryption, which therefore was to be kept private. We call them **private-key** or *symmetric* encryption schemes. Before using the scheme, the key must once be exchanged securely, hence private-key encryption is a "way of extending a private channel over time".

In the 1970s Diffie and Hellman [1] introduced a new concept, called **public-key** or *asymmetric* encryption: The encryption key differs from the decryption key, moreover, given the former it must be infeasible to find the latter. That is why these schemes provide secure communication without ever requiring any private channel; the receiver creates a pair of keys, gives the encryption key (that can be publicly known) to the sender, but keeps the decryption key secret. The sender can then use the encryption key to encrypt messages, which can only be decrypted by the receiver. In this context, the encryption and the decryption key are often referred to as the *public* and the *private key*, respectively.

An analogy illustrating the difference between the two notions of encryption is the problem of sending a confidential parcel by postal delivery: Private-key encryption corresponds to sending the secret content in a locked box. The drawback is that the recipient must be given the key to the box. A secure channel, e.g. a courier, is thus required. The idea to avoid this is not to send the key to the receiver, but rather let *him* send a *padlock* to the sender and keep the key. The sender locks the secret content and the receiver is the only one able to open the box. Intercepting the padlock is of no use to an adversary—assuming that it does not help in forging the key.

2. Probabilistic Public-Key Encryption

The first public-key cryptosystems (such as RSA [4]) were deterministic algorithms based on **trapdoor functions**. These are functions that are easy to compute but hard to invert—unless some information called the **trapdoor** is known. So, while everybody can use the function to encrypt messages, only the legal receiver knows the trapdoor, which serves as a decryption key.[3]

According to [2], the two main drawbacks of encryption schemes based on trapdoor functions are:

1. Inverting may be easy for plaintexts of some *special form.*[4]
2. It could be easy to compute at least *partial information* of the plaintext.

[2]More formally, probabilistic algorithms can be defined by Turing machines having an additional infinite read-only tape containing random bits.

[3]The trapdoor function used for RSA is exponentiation to the power of the public key in $\mathbb{Z}_n$, where $n = pq$ is the product of two large primes. That is, the encryption of a plaintext $m \in \mathbb{Z}_n^*$ is $c := m^e \bmod n$. The prime factors of n can be considered as the trapdoor.

[4]RSA, for example, always encrypts the messages 1 and 0 to themselves.

Furthermore, for deterministic schemes it is easy to detect if a message is sent twice. These points inspired the development of **probabilistic** public-key encryption schemes by Goldwasser and Micali [2]. They substituted the notion of trapdoor functions by what they introduced as (unapproximable) **trapdoor predicates**: A predicate B is trapdoor and unapproximable if anyone can *select* an x such that $B(x) = 0$ or y such that $B(y) = 1$, but only those who know the trapdoor information can, given z, *compute* the value of $B(z)$. Goldwasser and Micali used the predicate "is quadratic residue modulo composite n" (see Section 4.).

Their scheme uses bitwise encryption, which depends on a sequence of random bits. However, messages are always uniquely decryptable. Two properties are:

1. Decoding is easy for the legal receiver of a message, who knows the trapdoor information, but *provably* hard for an adversary.
2. No information about the plaintext can be obtained from the ciphertext by an adversary.

Definition 2 [Probabilistic Public-Key Bit-Encryption Scheme]. *A* ***probabilistic public-key bit-encryption*** *scheme* $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ *with* ***security parameter*** n *consists of:*

- $\mathcal{K}$*, the* ***key generator****: A probabilistic algorithm that on input* n *outputs a pair* (e, d)*, where* e *is the* ***public key*** *and* d *is the* ***private key****.*
- $\mathcal{E}$*, the* ***encryption function****, with three inputs: the public key* e*, the plaintext bit* $b \in \{0, 1\}$*, and a random string* r *of length* $p(n)$ *for some polynomial* $p(\cdot)$*. We will write* $\mathcal{E}_e(b, r)$*.*
- $\mathcal{D}$*, the* ***decryption function****, with two inputs: the private key* d *and the ciphertext* c*. Again, we will write* $\mathcal{D}_d(c)$*.*

Moreover, decryption of any encryption of a bit yields the encrypted bit, i.e.

$$\forall\, n \in \mathbb{N}\ \ \forall\, (e, d) \in \mathcal{K}(1^n)\ \ \forall\, b \in \{0, 1\}\ \ \forall\, r \in \{0, 1\}^{p(n)} : \mathcal{D}_d\big((\mathcal{E}_e(b, r))\big) = b$$

3. Semantic Security

A minimal requirement for an encryption scheme is that it must be impossible to retrieve an encrypted plaintext for anybody not knowing the decryption key. However, as already pointed out, this condition may be too weak—in some applications even partial information gained from the plaintext could endanger security. This is why we demand it to be "infeasible to learn *anything* about the plaintext from the ciphertext" or, in other words, "whatever an eavesdropper can compute about the cleartext given the ciphertext, he can also compute without the ciphertext" [2]. Schemes fulfilling this requirement, such as the Goldwasser-Micali scheme, are called **semantically secure**.

Since the actual definition of semantic security in [2] is rather technical, we settle for giving some intuitive notions.

Let M be the set of all possible messages, and for all $m \in M$, let p_m be the probability that m is sent. In a semantically secure cryptosystem, even if the adversary knows these probabilities, it must be hard for him to extract any information about messages from their encryption.

Let $f\colon M \to V$ be a function defined on M that represents such *information* about messages. Let v^M be a value for $f(m)$ that has maximal probability p^M to occur when m is chosen at random.[5] Consider the following two games: (Note that we assume that the adversary knows the public key e)

Game 1. Randomly pick $m \in M$ and ask the adversary to guess the value of $f(m)$ without telling him m. The best the adversary can do, is always guess v^M; thus, the probability of being right is p^M.

Game 2. Let the adversary choose a function f defined on M. Randomly pick $m \in M$. Compute an encryption of m and give it to the adversary. Now ask the adversary to guess $f(m)$.

Informally, a cryptosystem is semantically secure if the adversary cannot win Game 2 with higher probability than Game 1.

4. The Quadratic Residuosity Problem

In this section we present number-theoretical results that underlie the Goldwasser-Micali encryption scheme. For the sake of readability, proofs are not given here; the interested reader is referred to the appendix.

By $\mathbb{Z}_n^* := \{a \in \mathbb{N} \mid 1 \le a \le n-1 \wedge \gcd(a,n) = 1\}$ we denote the multiplicative group of $\mathbb{Z}_n$, i.e. all numbers less than n that have multiplicative inverses modulo n.

Definition 3 [Quadratic Residues]. *An element $a \in \mathbb{Z}_n^*$ is said to be a* ***quadratic residue*** *(*square*) modulo n if there exists an $x \in \mathbb{Z}_n^*$, such that $x^2 \equiv a \pmod n$. Every such x is called a* ***square root*** *of a modulo n. If no such x exists, then a is called a* ***quadratic non-residue*** *modulo n. We denote the set of all quadratic residues modulo n by Q_n, the set of all quadratic non-residues by $\overline{Q}_n$.*

Lemma 1. *Let p be a prime. Then $|Q_p| = |\overline{Q}_p| = \frac{1}{2}|\mathbb{Z}_p^*| = \frac{p-1}{2}$.*

Lemma 2. *Let p and q be odd primes, $n := pq$, $a \in \mathbb{Z}_n^*$. Then $a \in Q_n$ if and only if $a \in Q_p$ and $a \in Q_q$.* For $n = pq$, we have $|Q_n| = \frac{1}{4}(p-1)(q-1)$.

Definition 4 [Legendre Symbol]. *Let p be an odd prime, a an integer, s.t. $\gcd(a,p) = 1$. The* ***Legendre Symbol*** *is defined to be*

$$\left(\frac{a}{p}\right) = \begin{cases} 1, & \text{if} \quad a \in Q_p \\ -1, & \text{if} \quad a \in \overline{Q}_p \end{cases}$$

Lemma 3. *Let p be an odd prime, $a, b \in \mathbb{Z}_p^*$. Then $\left(\frac{a}{p}\right)\left(\frac{b}{p}\right) = \left(\frac{ab}{p}\right)$.*

The **Jacobi symbol** is an extension of the Legendre symbol for composite $n = pq$:

$$\left(\frac{a}{n}\right) := \left(\frac{a}{p}\right)\left(\frac{a}{q}\right)$$

[5] That is, v^M is a value in V with probability $p^M := \max_{v \in V}(\sum_{m \in f^{-1}(v)} p_m)$.

By definition, Lemma 3 obviously holds for composite n instead of p, too.

We now turn to the question of computing the Legendre (Jacobi) symbol. For prime p this can easily be done by the following criterion, which therefore yields an algorithm for deciding quadratic residuosity in $\mathbb{Z}_p^*$.

Proposition 1 [Euler's Criterion]. *Let p be an odd prime. Then $a \in \mathbb{Z}_p^*$ is a quadratic residue modulo p if and only if $a^{(p-1)/2} \equiv 1 \pmod{p}$*

For composite $n = pq$, there also exist efficient algorithms to compute the Jacobi symbol of a number a, even if the prime factorization of n is not known.

However, in contrast to the case $\mathbb{Z}_p^*$, this does *not* yield a tool for deciding quadratic residuosity in $\mathbb{Z}_n^*$: A quarter of the numbers in $\mathbb{Z}_n^*$ are quadratic residues while half of them have Jacobi symbol 1. The numbers having Jacobi symbol 1 while being quadratic non-residues are called **pseudosquares**.

Definition 5 [Quadratic Residuosity Problem (QRP)]. *Given a composite integer $n = pq$ and $a \in \mathbb{Z}_n^*$ with $(\frac{a}{n}) = 1$, decide whether or not a is a quadratic residue modulo n.*

There is no efficient procedure known for solving the Quadratic Residuosity Problem if the factorization of n is unknown. The **Quadratic Residuosity Assumption** states that for sufficiently large primes p and q, for every real-life algorithm it is infeasible to solve QRP.

However, if the factorization $n = pq$ is known, it is easy to solve QRP by computing $(\frac{a}{p})$, since a is a pseudosquare if and only if $(\frac{a}{p}) = (\frac{a}{q}) = -1$. It is these two facts that Goldwasser and Micali have based the first semantically secure cryptosystem upon.

5. The Goldwasser-Micali Encryption Scheme

We present the three algorithms $\mathcal{K}$, $\mathcal{E}$ and $\mathcal{D}$ of the Goldwasser-Micali encryption scheme as given in [3].

Algorithm 1 [Key Generation $\mathcal{K}$].

1. *Select two large random primes p and q, $p \neq q$.*
2. *Set $n \leftarrow pq$.*
3. *Select a pseudosquare $y \in \mathbb{Z}_n$ (i.e. y is quadratic non-residue and $(\frac{y}{n}) = 1$).*
4. *The public key is (n, y), the private key is (p, q).*

Algorithm 2 [Encryption $\mathcal{E}$]. *Let message m be a binary string $m = m_1 m_2 \ldots m_\ell$, let (n, y) be the public key.*

1. *For $i = 1 \ldots \ell$ do:*
   (a) *Select $x \in \mathbb{Z}_n^*$ at random.*
   (b) *If $m_i = 0$, set $c_i \leftarrow x^2 \bmod n$; otherwise set $c_i \leftarrow yx^2 \bmod n$.*
2. *The ciphertext is $c = (c_1, c_2, \ldots, c_\ell)$.*

Algorithm 3 [Decryption $\mathcal{D}$]. *Let $c = (c_1, c_2, \ldots, c_\ell)$ be a ciphertext and (p, q) the private key.*

1. *For $i = 1 \dots \ell$ do:*
   (a) *Compute $e_i = (\frac{c_i}{p})$ using Proposition 1.*
   (b) *If $e_i = 1$, set $m_i \leftarrow 0$; otherwise set $m_i \leftarrow 1$.*
2. *The decrypted message is $m = (m_1, m_2, \dots, m_\ell)$.*

Remark 1.

Key Generation: *The pseudosquare y required in Step 3 can be found by a probabilistic algorithm that picks y at random until $(\frac{y}{p}) = (\frac{y}{q}) = -1$.*

Encryption: *If $m_i = 0$, then it is encrypted to a random quadratic residue modulo n, while for $m_i = 1$ a random pseudosquare is chosen. In fact, according to Lemma 4 and multiplicativity of the Jacobi symbol, a pseudosquare times a quadratic residue yields a pseudosquare.*

Decryption: *Knowing the factors of n, it is easy to decide whether a c_i is a quadratic residue or not: $(\frac{c_i}{p}) = 1$ if and only if c_i is a quadratic residue, which is the case if and only if $m_i = 0$.*

Remark 2 [Security of the scheme]. *Assuming the hardness of QRP (Definition 5), the Goldwasser-Micali encryption scheme is semantically secure: For $x \in \mathbb{Z}_n^*$ is picked at random, x^2 is a random quadratic residue and yx^2 is a random pseudosquare modulo n. So, in order to decrypt a single bit of the ciphertext, an attacker would have to solve the quadratic residuosity problem. For a detailed proof based on mathematical definitions, see [2].*

The Goldwasser-Micali cryptosystem was the first system based upon the concept of probabilistic encryption and furthermore the first system proven to be semantically secure (assuming the intractability of the quadratic residuosity problem). It is, nevertheless, not a practicable scheme since in general, one plaintext-bit is expanded into n bits of ciphertext.

A Proofs

This appendix contains the proofs of the results stated in Section 4..

Lemma 4. *Let a be a quadratic residue modulo n, b a quadratic non-residue modulo n. Then ab is a quadratic non-residue modulo n.*

Proof. Since a is a quadratic residue, there exists $c \in \mathbb{Z}_n^*$ s.t. $c^2 \equiv a \pmod{n}$. Assume ab were a quadratic residue, too; let $d \in \mathbb{Z}_n^*$ be s.t. $d^2 \equiv ab \pmod{n}$. This implies $b \equiv d^2a^{-1} \equiv (dc^{-1})^2 \pmod{n}$. So dc^{-1} would be a root of b, contradicting its assumed non-residuosity. □

Proof.[Proof of Lemma 1] We show that for p prime, half of the elements of Z_p^* are quadratic residues.

Since $\mathbb{Z}_p^*$ is cyclic, it has a generator. Let g be s.t. $\mathbb{Z}_p^* = \{g^i \mid 0 \leq i \leq p-1\}$. We show that g is a quadratic non-residue modulo p: Assume that there is some

$a \in \mathbb{Z}_p^*$ s.t. $a^2 \equiv g \pmod p$. Thus $g^{(p-1)/2} \equiv a^{p-1} \equiv 1 \pmod p$ (by Fermat's Little Theorem[6]), which contradicts the fact that the order of the generator g is $p-1$.

$g^2, g^4, \ldots, g^{p-1}$ are distinct quadratic residues, while $g, gg^2, \ldots, gg^{p-1}$, being products of g and a quadratic residue, are quadratic non-residues by Lemma 4. □

Proof. [Proof of Lemma 2] We show that for $n = pq$, a is a quadratic residue modulo n if and only if it is a quadratic residue both modulo p and modulo q.
"⇒" Let $a \in Q_n$. There exists $c \in \mathbb{Z}_n^*$, s.t. $c^2 \equiv a \pmod n$. Therefore[7], $c^2 \equiv a \pmod p$; thus, c is a square root of a modulo p. Analogously, c is a square root of a modulo q.
"⇐" Let $a \in Q_p$ and $a \in Q_q$. There exist c_p and c_q, s.t. $c_p^2 \equiv a \pmod p$ and $c_q^2 \equiv a \pmod q$. By the Chinese Remainder Theorem there is one solution $x < n$ to the following system of congruences:

$$x \equiv c_p \pmod p$$
$$x \equiv c_q \pmod q$$

Thus $x^2 \equiv a \pmod p$ and $x^2 \equiv a \pmod q$, therefore $x^2 \equiv a \pmod n$.[8] □

Proof. [Proof of Lemma 3] We show that for $a, b \in \mathbb{Z}_p^*$, we have $(\frac{a}{p})(\frac{b}{p}) = (\frac{ab}{p})$. As in the proof of Lemma 1, let g be a generator of $\mathbb{Z}_p^*$ and let i be s.t. $a \equiv g^i \pmod p$. We know that $(\frac{a}{p}) = 1$ if and only if i is even. Let j be s.t. $b \equiv g^j \pmod p$. Thus $ab \equiv g^{i+j} \pmod p$ and $i+j$ is even if and only if neither or both of i and j are even. □

Proposition 2 [Euler's Criterion]. *Let p be an odd prime. Then for every $a \in \mathbb{Z}$*

$$\left(\frac{a}{p}\right) = 1 \quad \textit{if and only if} \quad a^{(p-1)/2} \equiv 1 \mod p$$

Proof. The map $\varphi\colon \mathbb{Z}_p^* \to \mathbb{Z}_p^*$, $a \mapsto a^{(p-1)/2}$ is a group homomorphism.[9] Let $\psi\colon \mathbb{Z}_p^* \to \{\pm 1\}$, $a \mapsto (\frac{a}{p})$. By Lemma 3, this is a group homomorphism, too. If $a \in \ker(\psi)$, then $a = c^2$ for some $c \in \mathbb{Z}_p^*$, so[10]

$$\varphi(a) = a^{(p-1)/2} = (c^2)^{(p-1)/2} = c^{p-1} = 1$$

by Fermat's Little Theorem. Thus $\ker(\psi) \subset \ker(\varphi)$. By Lemma 3, $\ker(\psi)$ has index 2 in $\mathbb{Z}_p^*$, i.e. it contains half of the elements of Z_p^*, so either $\ker(\varphi) = \ker(\psi)$ or $\varphi(a) = 1$ for all a. In the latter case, the polynomial $x^{(p-1)/2} - 1$ would have $p-1$ roots[11] in the field Z_n^*, so $\ker(\varphi) = \ker(\psi)$. □

[6] Fermat's Little Theorem states the following: For p prime and $a \in \mathbb{Z}_p^*$, we have $a^{p-1} \equiv 1 \pmod p$.

[7] Per definitionem $c^2 \equiv a \pmod n$ iff $n \mid c^2 - a$, so obviously $p \mid c^2 - a$.

[8] If two distinct primes p and q divide $x^2 - a$, then they both appear in the prime factorization of $x^2 - a$, thus also their product $pq = n$ divides $x^2 - a$.

[9] Since $\mathbb{Z}_p^*$ is an abelian group, we have: $\varphi(a)\,\varphi(b) = a^{(p-1)/2}\, b^{(p-1)/2} = (a\,b)^{(p-1)/2} = \varphi(a\,b)$.

[10] The kernel $\ker(\varphi)$ of a group homomorphism $\varphi\colon G \to H$ is a subgroup of the domain G containing exactly those elements that are mapped to the neutral element in H, i.e. $\ker(\psi) := \{a \in G \mid \varphi(a) = 1\}$.

[11] A basic result from field theory is the following: A polynomial of degree d over a field can have at most d roots.

Literatura

[1] W. DIFFIE, M. HELLMAN, New Directions in Cryptography, *IEEE Transactions on Informations Theory,* IT-22(6), pp. 644–654, 1976

[2] S. GOLDWASSER, S. MICALI, Probabilistic Encryption, *Journal of Computer and System Sciences*, 28, pp. 270–299, 1984

[3] A. J. MENEZES, P. C. VAN OORSCHOT, S. A. VANSTONE, *Handbook of Applied Cryptography,* CRC Press, Boca Raton, FL, 1996

[4] R. RIVEST, A. SHAMIR, L. ADLEMAN, A Method for Obtaining Digital Signature and Public Key Cryptosystems, *Communications of the ACM,* 21(2), pp. 120–126, 1978