# Making the Naccache-Stern Knapsack-type cryptosystem semantically secure via El-Gamal *

Santiago Egido†, Jorge Jiménez and Sebastià Martín
Dept. Matemàtica Aplicada IV. Universitat Politècnica de Catalunya
† CIMNE. Universitat Politècnica de Catalunya
Campus Nord, c/Jordi Girona, 1-3, 08034 Barcelona
e-mail: segido@cimne.upc.es, {jjimenez,sebasm}@ma4.upc.edu

**Abstract**

This paper describes a new cryptosystem generalising the Naccache-Stern knapsack (NSK) presented in [10]. Our main objective in this new construction was to equipe NSK with semantic security to provide the first knapsack-type cryptosystem with this level of security. The cryptosystem however could be considered also as in the line of El-Gamal-type schemes and, in this way, we can base semantic security on the difficulty of the Decisional Diffie-Hellman (DDH) assumption, one of the most popular and trustful assumptions in this context. This connection between this two type of cryptosystems can be reinterpreted as a connection between additive and multiplicative knapsacks, and it could be worthwhile to study it in the future.

**Keywords:** Public Key Cryptography, Knapsack Cryptosystems, Semantic Security, Discrete Logarithm.

## 1 Introduction

Since the renowned paper by R.C. Merkle and M. E. Hellmann [9] published in 1978, there have been many papers dedicated to design efficient and secure cryptosystems based on the difficulty of the subset sum problem or, as it is also known, knapsack problem. However, most of the subsequent systems based on this problem were systematically broken some time after their appearance, (see [2, 11] for an overview of knapsack cryptosystems and the attacks they have suffered). In 1982, Shamir [12]

*This work was partially supported by the Spanish *Ministerio de Ciencia y Tecnología* under project TIC 2003-00866.

broke the former knapsack system, as well as the hopes the cryptographic community had found in this efficient construction. So dramatic is the situation in knapsack cryptosystems that this problem has finally been removed from the standard specifications for public key cryptography [6]. Nevertheless, this kind of cryptosystems are interesting due to its efficiency and elegancy. Moreover it is known that, in general, the problem is NP-complete, providing a high level of confidence on its security, (although there have been some detractors too, see [11]).

For convenience we will include the definition of the subset sum problem in the next section. By now let us say that the main reason for the speed of this type of systems relies on the linearity of its mathematical structure. This is, on the other hand, one of its weaknesses. Indeed, one of the most important attacks to knapsack cryptosystems comes from the use of basis reduction algorithms ([12, 1, 8]).

To avoid this problem, Naccache and Stern [10] published in 1997 a multiplicative knapsack cryptosystem which, as far as we know, has not been attacked ever since. Despite of this fact, unfortunately, we can not trust the security of the Naccache-Stern cryptosystem, NSK from now on, since there is no proof supporting its security. The main purpose of this paper is to generalize NSK in such a way to obtain the same efficiency in encryption and decryption, but in a secure manner.

In 1984, Goldwasser and Micali [5] defined a new security notion that any encryption scheme should satisfy, namely, indistinguishability of encryptions or semantic security. This notion informally requires that a ciphertext should not leak any useful information about the plaintext, except its length, to a polynomial-time attacker. This security notion became a standard requirement for the design of new cryptosystems. To achieve this goal one exploits the fact that certain problems are considered hard to solve. Then the proof of security constructs a reduction map which, in case of breaking the cryptosystem provides an algorithm to solve this problem. One of the main problems used in cryptography to ensure security is the Discrete logarithm problem (DLP). The analysis about the intractability of the problem is so wide in literature that makes one feel quite safe when considering it as the "hidden" difficult problem of the cryptosystem. We will design our generalization based on DLP.

Knapsack cryptosystems in general, and in particular NSK, are not semantically secure. In this paper we present a new cryptosystem. It should be considered in the middle between Knapsack-type cryptosystems and El-Gamal type cryptosystems. In fact, as we have mentioned, it is constructed generalizing the multiplicative knapsack cryptosystem NSK, but in such a way that we can add security. In particular, its design, close to El-Gamal, allows us to build a security proof based on the so called Decisional Diffie-Hellman assumption (DDH) in the context of discrete logarithm problems, as it is the case in El-Gamal type cryptosystems. This cryptosystem does not intend to compete in practice with other El-Gamal type cryptosystems. In fact, in the best of the choices, the size of the keys needed is the same as the one achieved

in NSK; rather, the interest of our cryptosystem is of a more theoretical nature and comes in two different ways. On one hand, it provides a secure generalization of NSK cryptosystem; on the other hand, in our way to do so, we found an interesting connection between NSK, a multiplicative knapsack, and the more traditional additive knapsacks. In our opinion it would be interesting to consider how deep and strong this connection can be. In particular, can one hope to translate results about additive knapsacks into NSK? And, in view of the history of additive knapsacks, can one hope to break NSK with additive techniques via this new approach?

The paper is organized as follows. In Section 2 we include the preliminaries we need to develop the theory. Section 3 is dedicated to analyzing multiplicative knapsack cryptosystems, and to present our proposal. In Section 4, semantic security of the scheme is proven. To finish the paper we include some computations and examples in Section 5 and also an open problem which appeared to us during the implementation.

# 2 Preliminaries

## 2.1 Intractable problems

There are several ways to define knapsack problems. Merkle and Hellman started from the most common one for cryptographic purposes, namely:

**Definition 1** *Subset Sum Problem, SSP*

*Given a finite set of positive integers $A = \{a_1, \ldots, a_n\}$, called weights, and $M \in \mathbb{N}$, determine (provided it exists) a subset $I \subseteq \{1, \ldots, n\}$ such that $M = \sum_{i \in I} a_i$.*

As we have already mentioned, this is an $NP$-complete problem in terms of complexity. However there are many particular instantiations that are trivial to solve, as it is the case when the set $A$ is formed by a superincreasing sequence. To transform this problem into a cryptosystem, we use a difficult instantiation of the problem to encrypt, and then a trapdoor to obtain a trivial instantiation from the encryption. In the particular case of [9], to implement the system the authors needed to use a modular version of the problem which we state for convenience as follows:

**Definition 2** *Modular Subset Sum Problem (MSSP).*

*Given $p \in \mathbb{N}$, a finite set of integers $A = \{a_1, \ldots, a_n\}$, and $M \in \mathbb{N}$, determine (provided it exists) a subset $I \subseteq \{1, \ldots, n\}$ such that $M = \sum_{i \in I} a_i \bmod p$.*

For the rest of the paper we are interested in MSSP's, rather than SSP's.

As usual, to obtain a cryptosystem from a difficult problem, two additional conditions need to be fulfilled: firstly, we need to guarantee the existence of a unique plaintext for each valid ciphertext (uniqueness condition that we will denote

U), and secondly we need an easy decryption (D). Hence, in order to use an MSSP to design a cryptosystem, the set of weights has to verify, first of all, a condition which guarantees (U). Namely,

$$\sum_{i\in I} a_i \neq \sum_{j\in J} a_j \bmod p,\ \forall I, J \subseteq \{1,\dots,n\},\ I \neq J. \tag{1}$$

To achieve (D), the cryptosystem will use a trapdoor to convert a believed difficult instance of an MSSP into an easy one which allows decryption.

In Section 3 we will present a multiplicative analogous of the previous modular problem. The study of the hardness of our new proposal will lead us, in a natural way, to problems related to discrete logarithms, and in particular, as we mentioned in the introduction, the security of our cryptosystem relies in the DDH assumption, one of the most popular and trustful assumptions in this context.

There are many ways to formulate the Decisional Diffie-Hellman assumption, depending, among other things, on the precise probability space we are considering. Following the notation in [13],we will focus on a decisional problem of medium granularity. In other words, we will fix a group, and compute discrete logarithms with respect to a generator, chosen uniformly at random.

In particular, given a security parameter $\ell$, a prime $p$ chosen at random, and $G \leq \mathbb{Z}_p^*$ a subgroup of prime order $q$ such that $|q| = \ell$, then DDH assumes that there is no probabilistic polynomial time algorithm, in the security parameter $\ell$, which is able to distinguish the 4-tuple $(g,\ g^x,\ g^y,\ g^{xy})$ from $(g,\ g^x,\ g^y,\ g^z)$ with probability non-negligibly bigger than $1/2$, for $g$ chosen randomly among the generators in $G$, and $x, y, z$ random integers between 1 and $p-1$, (see [4]). In this sense, we can formulate DDH assumption in the following way:

**Assumption 3** *Decisional Diffie-Hellman Assumption (DDH).*

*Let $p$ be a given prime, $G \leq \mathbb{Z}_p^*$ a subgroup of prime order $q$, $g$ chosen uniformly at random among the generators of $G$ and $x, y, z$ random integers in the set $[1,\dots,p-1]$. Then, the probability distributions $D_{\mathrm{DH}} = (g,\ g^x,\ g^y,\ g^{xy})$ and $D_{\mathcal{R}} = (g,\ g^x,\ g^y,\ g^z)$ are polynomially indistinguishable.*

## 2.2 Previous knapsack cryptosystems

Our scheme is inspired in two previous knapsack cryptosystems that we include here for completeness. To simplify notation, for any given $n \in \mathbb{N}$, we will denote $\mathcal{I}_n = \{1,\dots,n\}$ and $\mathcal{I}_n^*$ the set of integers in $\mathcal{I}_n$ which are coprime to $n$.

The first knapsack cryptosystem is the one designed by Merkle and Hellmann [9], directly using the difficulty of the modular subset sum problem. In particular, to send an $n$ bit message through an insecure channel, the receiver considers a super-increasing sequence of positive integers, i.e. $a_1,\dots,a_n \in \mathbb{N}$ such that $a_i > \sum_{j=1}^{i-1} a_j$

$\forall i = 2, \ldots, n$ and a secret modulus $N > \sum_{i=1}^{n} a_i$. She then chooses at random a secret element $s \in \mathbb{Z}_N^*$, and publishes $c_i \equiv a_i s \pmod{N}$ for $i = 1, \ldots, n$. The encryption of an $n$-bit message $m = \sum_{i=1}^{n} m_i 2^{i-1}$ will be $C = \sum_{i=1}^{n} m_i c_i$. Decryption of the message is simple since $Cs^{-1} \equiv \sum_{i=1}^{n} m_i a_i \pmod{N} = \sum_{i=1}^{n} m_i a_i$, and $m_i$ are easily computable now, because the subset sum problem is constructed from a superincreasing sequence. Unfortunately it was proved that also the MSSP described by the sequence $\{c_i\}$ is easy to solve. One particular weakness arises from the fact that there could be several different pairs $(s', N')$ converting $\{c_i\}_i$ into a superincreasing sequence $\{b_i\}$, $b_i \equiv c_i s' \pmod{N'}$ $\forall i \in \{1, \ldots, n\}$, which makes possible to an attacker to recover the message. In this way, Shamir exploited in [12] the linearity of this knapsack cryptosystem, and efficiently computed a new pair $(s', N')$ in the conditions above, breaking the cryptosystem.

In order to avoid this attack, and inspired in some previous work, Naccache and Stern [10] built up a multiplicative version of the Merkle-Hellmann cryptosystem (we will call it NSK) that has not been broken so far.

To send an $n$-bit long message with NSK-cryptosystem the receiver, analogously to the former knapsack, considers the $n$ first primes, $p_i$, $i = 1 \ldots n$, and a prime $p > \prod_{i=1}^{n} p_i$. Then she chooses a secret integer $s \in \mathcal{I}_{p-1}^*$, at random and publishes $l_i = \sqrt[s]{p_i} \bmod p$, $i = 1, \ldots, n$ to be the weights for the hard knapsack used for encryption. The $n$-bit message $m = \sum_{i=1}^{n} m_i 2^{i-1}$ is now encrypted as $c = \prod_{i=1}^{n} \ell_i^{m_i} \bmod p$. The decryption is possible by computing $m = \sum_{i=1}^{n} \frac{2^{i-1}}{p_i - 1}(h_{i,s} - 1)$, where $h_{i,s} = \gcd(p_i, c^s \bmod p)$, by the choice of $p$ and the property of unique factorization in $\mathbb{Z}$. We point out that, although this scheme has not being broken, there is no result certifying its security.

# 3 Multiplicative knapsack cryptosystems

## 3.1 Subset product problems

The two preceding knapsacks are in fact very related. Indeed, if $p$ is a prime and $g$ a generator of $\mathbb{Z}_p^*$, for any $p_i$ we can write $p_i \equiv g^{\alpha_i} \bmod p$, for some integers $\alpha_i \in \mathcal{I}_{p-1}$ and then we see that,

$$\prod_{i=1}^{n} (p_i^{m_i} \bmod p) \equiv \prod_{i=1}^{n} (g^{\alpha_i m_i} \bmod p) \equiv g^{\sum_{i=1}^{n} \alpha_i m_i} \bmod p\,,$$

translating the multiplicative problem in the Naccache-Stern cryptosystem into a subset sum problem at the exponent. In this sense, we can deduce the necessary conditions to design a multiplicative knapsack cryptosystem from those of the additive problem. In particular, the public key in an additive knapsack consists of the weights of a hard subset sum problem. It is then natural to think about the multiplicative version of this difficult problem.

**Definition 4** *Modular Subset Product Problem, MSPP*

*Given $p \in \mathbb{N}$, a finite set of integers, $T = \{t_1, \ldots, t_n\}$, coprime to $p$, and $S \in \mathbb{N}$, determine (provided it exists) a subset $I \subseteq \{1, \ldots, n\}$ such that $S \equiv \prod_{i \in I} t_i \bmod p$.*

Observe that if $p > \max(S, \prod_{i=1}^{n} t_i)$ and the integers in $T$ are pairwise coprime, then the problem has polynomial complexity (in fact quadratic, by using Euclid's algorithm). In general, by the observation above, it is clear that the hardness of this problem is closely related to problems in the discrete logarithm context, as we will see below. In this way, this formulation in multiplicative terms will be useful to design semantically secure knapsack cryptosystems. For this purpose we need Conditions (U) and (D) to be fulfilled. The multiplicative analogous of (1) is

$$\prod_{i \in I} t_i \neq \prod_{j \in J} t_j \bmod p, \ \forall I, J \subseteq \{1, \ldots, n\}, \ I \neq J. \tag{2}$$

In the case of NSK, Condition 2 is achieved by choosing $t_i = p_i$ for $i = 1, \ldots, n$, the $n$ first primes, due to uniqueness of factorization, for $p$ large enough. In this sense they needed

$$p > \prod_{i=1}^{n} t_i = \prod_{i=1}^{n} \left(g^{\alpha_i} \bmod p\right). \tag{3}$$

On the other hand, Condition (D) in the case of NSK is a consequence of Euclid's algorithm.

**Example 5** *A new Knapsack cryptosystem based on a MSPP.*

To build up a new multiplicative knapsack cryptosystems we just have to find a family $T = \{t_i\}_i$ fulfilling Conditions 2 and 3. A trivial choice for this purpose would be $t_i = 2^{2^{i-1}}$, for $i = 1, \ldots, n$, $p > 2^{2^n}$, which trivially verify (U) exactly as the superincreasing sequence does in the additive case. Observe that the size of the keys is however exponentially bigger in the multiplicative problem than in the additive one. From here, we can design the cryptosystem in the following way. We choose a secret key $s$ coprime to $p - 1$. Then we publish $p$ and $v_i = t_i^d \bmod p$, $i = 1, \ldots, n$, where $ds = 1 \bmod p - 1$. The message $m = \sum_{i=1}^{n} m_i 2^{i-1}$ is encrypted as $c = \prod_{i=1}^{n} v_i^{m_i} \bmod p$, and then decrypted by the receiver computing $c^s \bmod p = \prod_{i=1}^{n} 2^{2^{i-1} m_i}$, and solving a trivial additive subset sum problem.

This example is just the translation of choosing a superincreasing sequence in the additive case. However, in this multiplicative context it is natural to think that there could be better alternatives of choosing the weights in $T$ ensuring Condition 2, by exploiting multiplicative properties of the bases instead of the size of the exponents. In this way we can choose any selection of pairwise coprime integers $t_i$ in the MSPP, which in fact include the selection in NSK. In any event, the security of a cryptosystem in this general family still would need to be proved.

### 3.2 Semantically secure knapsack

We now present our proposal of semantically secure knapsack cryptosystem. As we mentioned above, semantic security will be based on the DDH assumption.

Let $n$ be an integer, $\ell$ a security parameter, $p$ a prime, $G \leq \mathbb{Z}_p^*$ such that $|G| = q$ is a prime of length $\ell$, $g \in G$ a generator, and $T = \{t_1, \ldots, t_n\} \subset G$ an easy instance of an MSPP satisfying Conditions (2) and (3) , where $\alpha_i \in \mathcal{I}_q$, $i = 2, \ldots, n$, and $\alpha_1 \in \mathcal{I}_q^*$.
The parameters of the cryptosystem with set of plaintexts $\mathcal{M}_n$, the set of integers of $n$ bits, and message $m = \sum_{i=1}^n m_i 2^{i-1}$ expressed on base 2, are as follows:

- *Secret key*: a random integer $s \in \mathcal{I}_q^*$.
- *Public key*: $p$, $v_1 = {t_1}^d \bmod p$, and $\ell_i = t_i^{d^2} \bmod p$, for $i = 1, \ldots, n$, where $d = s^{-1} \bmod q$.
- *Encryption*: Choose at random $b \leftarrow \mathcal{I}_{p-1}$. The ciphertext is $c(m, b) = (c_1, c_2)$, with $c_1 = v_1^b \bmod p$, $c_2 = \ell_1^b \prod_{i=1}^n \ell_i^{m_i} \bmod p$.
- *Decryption*: Compute $c_1^{-s} c_2^{s^2} = \prod_{i=1}^n g^{\alpha_i m_i} \equiv \prod_{i=1}^n t_i^{m_i} \bmod p$.

We point out that the ciphertext can be represented in terms of the generator as $c(m, b) = (g^{db\alpha_1}, g^{d^2(b\alpha_1 + \sum_{i=1}^n \alpha_i m_i)})$. On the other hand, the ciphertext in NSK was only $\prod_{i=1}^n \ell_i^{m_i}$ and, since no randomness was considered, there was not any chance for the scheme to be semantically secure.

# 4 Semantic security

In this section we prove that our knapsack is semantically secure (IND–CPA) in the standard model, under DDH assumption on a subgroup $G$ of $\mathbb{Z}_p^*$, for a given $p$ prime, and a generator $g \in G$ varying at random (i.e. medium granularity, according to [13]). To get this level of security we need to ensure that the probability that an adversary obtains some information from the ciphertext, which he could not achieve from the public data, is negligible. If we denote $\mathcal{D}_0$ the probability distribution associated to the set of encryptions $c(m_0, b)$ for a fixed message $m_0 \in \mathcal{M}_n$, and $\mathcal{D}$ the probability distribution associated to the set of encryptions $c(m, b)$ for a random message $m \leftarrow \mathcal{M}_n$, a standard way to prove semantic security is to show that, $\mathcal{D}_0$ and $\mathcal{D}$ are polynomially indistinguishable distributions for any fixed $m_0 = \sum_{i=1}^n m_{i,0} 2^{i-1}$. From now on we will denote by $D_1 \approx D_2$ the fact that two probability distributions $D_1$ and $D_2$ are polynomially indistinguishable [4].

In our case, the public information is gathered into $v_1$ and $l_i$, $i = 1 \ldots, n$, and therefore

$$\mathcal{D}_0 = (g^{d\alpha_1}, g^{d^2\alpha_1}, \ldots, g^{d^2\alpha_n}, g^{db\alpha_1}, g^{d^2(b\alpha_1 + \sum_{i=1}^n \alpha_i m_{i,0})}),$$
$$\mathcal{D} = (g^{d\alpha_1}, g^{d^2\alpha_1}, \ldots, g^{d^2\alpha_n}, g^{db\alpha_1}, g^{d^2(b\alpha_1 + \sum_{i=1}^n \alpha_i m_i)}),$$

where $b \leftarrow \mathcal{I}_{p-1}$, $m \leftarrow \mathcal{M}_n$, $d \leftarrow \mathcal{I}_q^*$ and $g \in G$ is a given generator.

The following lemma will allow us to simplify dramatically the proof of semantic security.

**Lemma 6** *Let $p = 1 + qr$ be a fixed prime, $\alpha_1, \ldots, \alpha_n \in \mathbb{N}$ such that $\alpha_1 \in \mathcal{I}_q^*$, and $X = \{(a^{\alpha_1}, b^{\alpha_1}, b^{\alpha_2}, \ldots, b^{\alpha_n}, c, d) \,|\, a, b, c, d \in G\}$. Consider the following map:*

$$\varphi : \quad X \quad \longrightarrow \quad G \times G \times G \times G$$
$$x \quad \rightarrow \quad (a, b, c, d)$$

*Then, $\varphi$ and $\varphi^{-1}$ are bijections that can be computed in polynomial time.*

*Proof*: bijectivity is trivial, and the only computations involved are modular exponentiations modulo $p$; namely 2 to compute $\varphi(x)$ and $n+1$ to compute $\varphi^{-1}(a, b, c, d)$. ■

Hence, proving $\mathcal{D}_0 \approx \mathcal{D}$ is equivalent to proving that the distributions

$$(g^d, g^{d^2}, g^{db\alpha_1}, g^{d^2(b\alpha_1 + \sum_{i=1}^n \alpha_i m_{i,0})}) \text{ and } (g^d, g^{d^2}, g^{db\alpha_1}, g^{d^2(b\alpha_1 + \sum_{i=1}^n \alpha_i m_i)})$$

are indistinguishable for $d \leftarrow \mathcal{I}_q^*$, $b \leftarrow \mathcal{I}_{p-1}$ and $m = \sum_{i=1}^n m_i 2^{i-1} \leftarrow \mathcal{M}_n$. Indeed, we just have to observe that for any bijection $\varphi$, such that $\varphi$ and $\varphi^{-1}$ can be computed in polynomial time, then $D_1 \approx D_2$ is equivalent to $\varphi(D_1) \approx \varphi(D_2)$.

**Theorem 7** *The proposed scheme is semantically secure if and only if DDH holds.*

*Proof*:

It is clear that, for any $\beta \in \mathcal{I}_{p-1}^*$ and any given plaintext $m_0 = \sum_{i=1}^n m_{i,0} 2^{i-1}$, the map

$$\theta : \quad G \times G \times G \times G \quad \longrightarrow \quad G \times G \times G \times G$$
$$(c_1, c_2, c_3, c_4) \quad \rightarrow \quad (c_1, c_2, c_3^{\beta_1}, c_4^{\beta_1} c_2^{-\beta_1 \sum_{i=1}^n \alpha_i m_{i,0}})$$

is a bijection that can be computed in polynomial time, and so, choosing $\beta_1 = \alpha_1^{-1} \bmod p - 1$, the problem is reduced to prove the indistinguishability of $D_0 = (g^d, g^{d^2}, g^{db}, g^{d^2 b})$, and
$D = (g^d, g^{d^2}, g^{db}, g^{d^2(b + \beta_1 \sum_{i=1}^n \alpha_i (m_i - m_{i,0}))})$, where $b \leftarrow \mathcal{I}_{p-1}$, $m \leftarrow \mathcal{M}_n$ and $d \leftarrow \mathcal{I}_q^*$.

Next reduction will essentially solve the problem. In this case we note that as $d$ varies over $\mathcal{I}_q^*$, $g^d$ runs through all the generators of $G$. Hence

$$D_0 = (g, g^d, g^b, g^{db}) \text{ and } D = (g, g^d, g^b, g^{d(b+\beta_1 \sum_{i=1}^n \alpha_i (m_i - m_{i,0}))}),$$

where $b \leftarrow \mathcal{I}_{p-1}$, $m \leftarrow \mathcal{M}_n$, $d \leftarrow \mathcal{I}_q^*$ and $g \leftarrow G$ generator.

Now let us see that, if DDH holds, then for any set $X \subset \mathcal{I}_{p-1}$ we have

$$(g, g^d, g^b, g^{d(b+x)}) \approx (g, g^d, g^b, g^z)$$

with the parameters $b$, $d$, $g$ in the respective sets, and $z \leftarrow \mathcal{I}_{p-1}$, $x \leftarrow X$. Indeed, suppose that both distributions are distinguishable. Then, given a DDH tuple $(c_1, c_2, c_3, c_4)$, we choose $x \leftarrow X$, and compute $(c_1, c_2, c_3, c_4 c_2^x)$. Note that this will be one of our 4-tuple, since $x \in I_p$ is a multiple of $q$ with negligible probability in the parameter $\ell$. In this way, to break DDH it is enough to distinguish $(g, g^d, g^b, g^{d(b+x)})$ from $(g, g^d, g^b, g^{z+dx})$. The proof finishes by observing that $z + dx \leftarrow \mathcal{I}_{p-1}$ independently of $X$ since, for given $k \in \mathcal{I}_{p-1}$ and a pair $(x, d)$, there exists a unique $z$ such that $z + dx = k$.

■

# 5 The system in practice

Let us see how to effectively build a particular semantically secure knapsack cryptosystem following the general description in Section 3.2. For this purpose we need to find an easy instantiation of MSPP given by a set $T = \{t_1, \ldots, t_n\} \subset \mathbb{N}$ satisfying Conditions (2) and (3), and such that $\alpha_1$ is coprime to $p - 1$, (in fact coprime to $q$ for some $q|p - 1$).

Let $k, m \in \mathbb{N}$, $T = \{2, 2^2, 2^4 \ldots 2^{2^k}, 3, 3^2, 3^4 \ldots, 3^{2^m}\}$. If we choose $g = 2$, then $\alpha_1 = 1$ and so $\alpha_1 \in \mathcal{I}_q^*$ for any $q$. It is important to note that 2 will in fact be a generator of $\mathbb{Z}_p^*$, with conjectural probability bigger than 1/3 (see [3]). Also observe that, in those cases, if $p = 1 + qr$, then $2^r$ will be a generator of a group $G$ of order $q$. If $p > 2^{2^{k+1}-1} 3^{2^{m+1}-1}$ then Condition (3) is satisfied. On the other hand, Condition (2) is trivially satisfied. Nevertheless, in the proof of semantic security we have used that the bijections in Lemma 6 are computable in polynomial time, which means we need to know the set $A = \{\alpha_1, \ldots, \alpha_n\}$. This can be guaranteed, for instance, by choosing $p$ among the divisors of $2^l - 3$, when $\ell$ runs through $\mathbb{N}$, because in this case computing $\log_2 3$ in $\mathbb{Z}_p$ is trivial. This observation would in fact be useful only if there are infinitely many primes dividing $2^\ell - 3$ for $\ell$ varying over the integers. This is guaranteed by the following lemma.

**Lemma 8** *The sets $P_\pm = \{p, prime : p|(b \pm 1)^k - b$ for some $k \in \mathbb{N}\}$ have infinitely many primes for any given integer $b$.*

*Proof*: Let $Q = \{p_1, \ldots, p_r\} \subset P_\pm$, and $a_l = (b \pm 1)^{1+l(p_1-1)\cdots(p_r-1)} - b, l \in \mathbb{N}$. Then, $a_l \equiv \pm 1 \bmod p_i$ for $i = 1, \ldots, r$, and therefore if $p|a_l$ then $p \in P_\pm \setminus Q$. ∎

As an aside, it is likely that there are infinitely many primes of the form $p = 2^a - 3^b$. To argue this, let $k$ be a large integer and consider the set $\{2^a - 3^b \mid a \in [k, 2k], b \in [1, k]\}$. All these numbers are smaller than $2^{2k}$, and therefore the probability that any of them is prime is larger than $1/2k$. Since we are considering $k^2$ numbers, we can hope that set to contain some $k/2$ primes. It is also likely that 2 or 3 will be generators for many of these primes. We note also that every prime $p$ will eventually divide a number of the form $2^a - 3^b$. There exist at least two primes, 683 and 599479, with the unusual property that all the non-one modular powers of 2 are different to all the non-one modular powers of 3; thus, the smallest nontrivial multiple of $p$ turns out to be $2^{p-1} - 3^{p-1}$. This is quite rare, though; a typical prime $p$ can be expected to have a multiple smaller than $3^{\sqrt{p}}$, since the set $\{2^a - 3^b \mid 1 \leq a, b \leq \sqrt{p}\}$ contains $p$ numbers with absolute value smaller than $3^{\sqrt{p}}$.

The generality of the previous lemma allows us to find quickly a prime $p$ and a generator $b$ of $\mathbb{Z}_p^*$ (by trying a few values $b$). We observe that, if $t_i = b^{2^i}$, $i = 1, \ldots, n$, then $l_{i+1} = l_i^2$, and we lose some randomness. However, we could take the set $\{b^{e_1}, b^{e_2}, \ldots, b^{e_k}, (b+1)^{f_1}, (b+1)^{f_2} \ldots (b+1)^{f_m}\}$ for any $\{e_i\}_i$, $\{f_i\}_i$ superincreasing sequences and $p$ a prime satisfying the corresponding Condition 3. We now show in full detail a simple instance with messages in $\mathcal{M}_{11}$. For this purpose we run a MAPLE program which lists primes dividing $2^h - 3^r$ for $1 \leq h \leq r \leq 100$. This enable us to find $\log_2(3) \bmod p$. Among those primes we need to choose the ones satisfying Condition 3 and, for that reason, we need first to determine the number of powers of 2 and 3 that we will include in our list. When $T = \{2, 2^2, 2^4 \ldots 2^{2^k}, 3, 3^2, 3^4 \ldots, 3^{2^m}\}$, it is easy to see that we will be able to use a smaller $p$ by choosing $m = k$ or $m = k - 1$. Considering $h = 27$, $r = 71$, then $p = 3^{71} - 2^{27}$ is a prime greater than $2^{63}3^{31}$, and so we can select $T = \{2, \cdots, 2^{32}, 3, \cdots, 3^{16}\}$ to be the set in our easy MSPP. We randomly select $s = 254486354087853147756367633915608 7$, coprime to $p - 1$ and compute $d = 485049474878014287802000224192679 9$. The set of weights will be $v_1 = 2^d \bmod p$ and $l_i = t_i^{d^2} \bmod p, i = 1 \ldots, n$. To encrypt the message $m = 1$, we randomly select $b = 688954424245652167225425748684388 9$, and send

$$\begin{aligned} c_1 &= 68025339141513491136080666696473 40, \\ c_2 &= 51454479692323499787264108451456 41. \end{aligned}$$

We see that $c_1^{-s} c_2^{s^2} \bmod p = 2$, from where we easily recover the message $m = 1$.

To encrypt the message $m = 11111111111$ we make a new random selection $b = 21428524262670563619304434955451 81$, and send

$$\begin{aligned} c_1 &= 31063676071868371765837460184419 46, \\ c_2 &= 37674478464591616507968888101662 66. \end{aligned}$$

We see that $c_1^{-s} c_2^{s^2} \bmod p = 2^{63} \cdot 3^{31}$, from where we easily recover the message $m$.

Clearly, the cryptosystem built in the previous way is highly inefficient. In particular, to encrypt 11 bits of plaintext, a 10 times more length ciphertext is needed (in NSK would approximately be 5.5 times). The main reason for this problem is Condition 3 to ensure decryption by factorization. The double exponential increasing at the original sequence in $T$ makes this selection unfeasible.

## 5.1 General instances

In general, we want Conditions 2 and 3 to be fulfilled, in order to make the system compatible with an easy decryption. There are many ways of achieving these properties. In particular, selecting any set $T$ of pairwise coprime integers, and a prime $p$ bigger than the product of the $t_i$. One possible choice is letting $T$ be the set of the $n$ first primes, as in [10]. In that case, the authors already showed the asymptotic growth of $p$ as $p \sim n! \log n^n e^{-n/\log n}$. In particular, for $n = 160$, a prime of about 900 bits is needed. The obstacle in these choices would be that, in our proof of semantic security, the knowledge of the exponents $\alpha_i$ is used in a decisive way. In this direction we address the following open problem, generalizing somehow the discrete logarithm problem.

**Problem 9** *Given integers $t_1, \ldots, t_n$, find $\alpha_1, \ldots, \alpha_n$, a prime $p$, and a generator $g \in \mathbb{Z}_p^*$ such that $g^{\alpha_i} \equiv t_i \bmod p$ for $i = 1 \ldots, n$, but such that DL is a hard problem in $\mathbb{Z}_p^*$.*

This obstacle can be avoided by a more flexible definition of semantic security. We observe that, if there exists an oracle such that given an instance of the scheme, it produces the exponents $\alpha_i$, then the scheme becomes semantically secure. In this direction we make the following definition:

**Definition 10** *A scheme $(P_k, S_k, E_k(m, r) = c, D_k(c) = m)$ is potentially semantically secure, PSS, if there exists an oracle $\mathcal{O}$ such that the scheme given by $(P_k \cup \mathcal{O}, S_k, E_k(m, r) = c, D_k(c) = m)$ is semantically secure.*

The idea is that even when we add more public information to a PSS scheme, it remains IND-CPA secure. In our case, the information we need to add with the help of the oracle $\mathcal{O}$ would be the exponents $\alpha_i$. The main observation is that, in some cases, PSS also implies semantic security.

We make the following assumption:

**Assumption 11** *Let $p$ be prime, and $q|p-1$ a prime of, at least, $n$ bits. Then $(g, g^e, \ldots, g^{e^n}) \approx (g, g^{\alpha_1}, \ldots, g^{\alpha_n})$ where $g, e$ are random generators of $\mathbb{Z}_p^*$ and $\mathbb{Z}_q^*$ respectively, and $\alpha_i \leftarrow \mathbb{Z}_q^*$, for $i = 1, \ldots, n$.*

The particular case of $n = 2$ of the previous assumption is the well known Decisional Square Exponent assumption, DSE, (see [13]).

Now, suppose there exists an adversary $\mathcal{A}$ who is able to break the semantic security of the cryptosystem. We build instances of the scheme as in the example in Section 5, choosing $T = \{b^e, \ldots, b^{e^n}\}$ for some integers $b$ and $e$ and a prime $p$ satisfying Condition 3. Hence, under assumption 11, the instantiations we can make with the previous selections of $T$ match perfectly the real setting. So the adversary cannot tell the difference, and the semantic secure challenge is the same as in the general case. Therefore, if we gave the public information, the challenge messages $m_0$ and $m_1$, and the ciphertext to the adversary $\mathcal{A}$; with its answer and the exponents, we would be able to distinguish Diffie-Hellman by following our proof of security.

### 5.2 Example

Following the tradition we include a challenge and offer an impressive collection of the, nowadays, ancient pesetas, to whom cryptanalyses the proposed scheme. We set $n = 95$, so we will be sending 96 bits length messages. Choose $t_i = p_i$ the $i$-th prime. The prime modulus is

$$\begin{aligned} p = \; & 964274047248418797145090983157197980855078966882276 4 \\ & 925727885329549041126553384393613062138985695165937 4 \\ & 426739175403330646512591919969270332387855783302357 33 \\ & 126850026706628464775925976598261134606198152447213 11. \end{aligned}$$

The list of weights is available sending an email to the authors. The cipher would be

$$\begin{aligned} c1 = \; & 673966455022394929975326107302312474023253501637825 11 \\ & 726496980399713572866396612846910458118307479909662 06 \\ & 020009716908682690109515650423920921183615717329632 16 \\ & 971936824799904962146207163271105372406635167698099. \end{aligned}$$

$$\begin{aligned} c2 = \; & 149414751045170737688589049123047625660755048823808 49 \\ & 883075059107125072664640827023479902647267483193018 92 \\ & 250676773820727101146097297113155430302826923678636 00 \\ & 051733248473026902132495959530369490728841903982637. \end{aligned}$$

The challenger should be the first to decrypt at least a 50% of this ascii coded message with a method that should be applicable to any length of the incoming message.

## 6 Concluding remarks

As we have seen, the security of this new cryptosystem is based on DLP. In this problem, as in the integer factorization, the fact that, in spite of the uncountable number of bibliographic references dealing with the problem of computing the

discrete logarithm in polynomial time, does not exist such an algorithm, makes the cryptosystems basing their security in this problem somehow more secure than those based on new ad-hoc asssumptions not so studied in the literature. On the other hand, as it is pointed out in [7], it would be a weakness to base all the cryptosystems on DLP or the integer factorization problem and, in this way, problems like the Vector Decomposition Problem (VDP) (see [14]), or the Nearest code word problem (NCP) appear. We thank an anonymous referee for pointing out the reference [7] which led to the following discussion.

It is our objective to show now that subset sum problems and VDC or NCP are slightly related. This would somehow add some interest in comparing multiplicative and additive knapsacks. Indeed, first we will see that SSP is equivalent to the following $n$-dimensional version of it.

**Definition 12** *Vectorial subset sum problem, VSSP*

*Given a finite set of elements in* $\mathbb{N}^{r+1}$, $\mathbf{A} = \{\mathbf{a_1}, \ldots, \mathbf{a_n}\}$, *called weights, and a vector* $\mathbf{M} \in \mathbb{N}^{r+1}$, *determine (provided it exists) a subset* $I \subseteq \{1, \ldots, n\}$ *such that* $\mathbf{M} = \sum_{i \in I} \mathbf{a}_i$.

Assuming $r$ is polynomial in $n$ and the size of $\mathbf{A}$, $|A| = \max\{a_{i,j}\}$, we have the following:

**Lemma 13** *There exists a polynomial time algorithm which reduces VSSP to SSP, and viceversa.*

*Proof*: We want to reduce VSSP to SSP. Let $\mathbf{M} = (M_1, \ldots, M_r)$ and $\mathbf{a}_i = (a_{i,0}, \ldots, a_{i,r})$. Given an integer $k > n|A| \cdot \max\{M_i\}$, let us consider the weights $b_j = 10^{jk} \sum_{i=1}^{r} a_{i,j}$ and the integer $M = \sum_{i=0}^{r} M_i 10^{ik}$. Then a solution $I \subseteq \{1, \ldots, n\}$ to $M = \sum_{i \in I} b_i$ is also a solution to $\mathbf{M} = \sum_{i \in I} \mathbf{a}_i$. In fact, the opposite is also true. The converse is straightforward.

Observe that VDP or NCP imply VSSP trivially. In the first case we are finding the coordinates of a vector $\mathbf{M}$ in the system $\mathbf{A}$ (and it happen to be 0 or 1 any of them), while n the second, VSSP is the particular case in which the word $\mathbf{M}$ is in fact part of the code. In this way, if VSSP is difficult, VDP or NCP will also be difficult.

# References


[1] E. F. Brickell, Solving low density knapsacks, *Adv. in Cryptology. Proc. Crypto'83* Plenum Press, pp. 25-37 (1984).

[2] Y.Desmedt, What happened with knapsack cryptographic schemes, *Applied Sciences* **142**, Kluwer Acad. Pub., pp. 113–134 (1988).

[3] P. D. T. A. Elliott and L. Murata, On the average of the least primitive root modulo *p. J. London Math. Soc. (2)* **56.3**, pp. 435–454. (1997).

[4] O. Goldreich, Foundations of cryptography-Basic tools. Cambridge Univ. Press. (2001).

[5] S. Golwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences* **28**, pp. 270–299 (1984).

[6] IEEE, P1363: Standard specifications for public key cryptography http://grouper.ieee.org/groups/1363/

[7] K. Kobara and H. Imai. Semantically Secure McEliece Public-Key Cryptosystems Conversions for McEliece PKC.*Proc. of PKC 01*, LNCS 1992, pages 1935. SpringerVerlag, 2001.

[8] J. C. Lagarias, A. M. Odlyzko. Solving low density subset sum problems. *J. Assoc. Comp. Mach.***32** Ed. C. Pomerance, Am. Math. Soc., pp. 229–246 (1985). preliminary version in *Proc. 24th IEEE Symposium on Foundations of Computer Science*, IEEE, pp. 1–10 (1983).

[9] R. Merkle and M. Hellman, Hiding information and signatures in trapdoor knapsacks, *IEEE Transactions on Information Theory* **24**, 5, pp. 525–530 (1978).

[10] D.Naccache and J.Stern, A new public key cryptosystem, *Advances in Cryptology: Proceedings of Eurocrypt'97*, pp. 27–36 (1997).

[11] A. M. Odlyzko, The rise and fall of knapsack cryptosystems, *Cryptology and Computational Number Theory*. Proc. Symp. Appl. Math., **42** Ed. C. Pomerance, Am. Math. Soc., pp. 75–88 (1990).

[12] A. Shamir, A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystem, *Advances in Cryptology-CRYPTO '82*.

[13] A. Sadeghi and M. Steiner, Assumptions related to discrete logarithms: why subtleties make a real difference, *Advances in Cryptology: Proceedings of Eurocrypt'01*, pp. 244–261 (2001).

[14] M. Yoshida, S. Mitsunari, and T. Fujiwara, Vector Decomposition Problem and the Trapdoor Inseparable Multiplex Transmission Scheme Based the Problem, Proceedings of the 2003 Symposium on Cryptography and Information Security (SCIS2003), pp.491-496 (2003-01).