Exploring Naccache-Stern Knapsack Encryption

Éric Brier[1], Rémi Géraud[2], and David Naccache[2]

[1] Ingenico Terminals
9 Avenue de la Gare F-26300 Alixan, France
eric.brier@ingenico.com
[2] École normale supérieure
45 rue d'Ulm, F-75230 Paris CEDEX 05, France
{remi.geraud,david.naccache}@ens.fr

Abstract. The Naccache–Stern public-key cryptosystem (NS) relies on the conjectured hardness of the modular multiplicative knapsack problem: Given $p, \{v_i\}, \prod v_i^{m_i} \bmod p$, find the $\{m_i\}$.

Given this scheme's algebraic structure it is interesting to systematically explore its variants and generalizations. In particular it might be useful to enhance NS with features such as semantic security, re-randomizability or an extension to higher-residues.

This paper addresses these questions and proposes several such variants.

1 Introduction

In 1997, Naccache and Stern (NS, [15]) presented a public-key cryptosystem based on the conjectured hardness of the modular multiplicative knapsack problem. This problem is defined as follows:

Let p be a modulus[3] and let $v_0, \ldots, v_{n-1} \in \mathbb{Z}_p$.

$$\text{Given } p, v_0, \ldots, v_{n-1}, \text{ and } \prod_{i=0}^{n-1} v_i^{m_i} \bmod p, \text{ find the } \{m_i\}.$$

Given this scheme's algebraic structure it is interesting to determine if variants and generalizations can add to NS features such as semantic security, re-randomizability or extend it to operate on higher-residues.

This paper addresses these questions and explores several such variants.

1.1 The Original Naccache–Stern Cryptosystem

The NS cryptosystem uses the following sub-algorithms:

[3] p is usually prime but nothing prevents extending the problem to composite RSA moduli.

– Setup: Pick a large prime p and a positive integer n.
Let $\mathfrak{P} = \{p_0 = 2, \ldots, p_{n-1}\}$ be the set of the n first primes, so that

$$\prod_{i=0}^{n-1} p_i < p$$

(We leave aside a one-bit leakage dealt with in [15] — this technique applies *mutatis mutandis* to the algorithm presented in this paper).

– KeyGen: Pick a secret integer $s < p - 1$, such that $\gcd(p - 1, s) = 1$. Set

$$v_i = \sqrt[s]{p_i} \bmod p.$$

The public key is $(p, n, v_0, \ldots, v_{n-1})$. The private key is s.

– Encrypt: To encrypt an n-bit message m, compute the ciphertext c:

$$c = \prod_{i=0}^{n-1} v_i^{m_i} \bmod p$$

where m_i is the i-th bit of m.

– Decrypt: To decrypt c, compute

$$m = \sum_{i=0}^{n-1} 2^i \mu_i(c, s, p)$$

where $\mu_i(c, s, p) \in \{0, 1\}$ is the function defined by:

$$\mu_i(c, s, p) = \frac{\gcd(p_i, c^s \bmod p) - 1}{p_i - 1}.$$

To this day, NS has neither been proven secure in the usual models, nor has it been attacked. Rather, its security relies on the conjectured hardness of a multiplicative variant of the knapsack problem[4]:

Definition 1 (Multiplicative Knapsack Problem). *Given p, c, and a set $\{v_i\}$, find a binary vector x such that*

$$c = \prod_{i=0}^{n-1} v_i^{x_i} \bmod p.$$

Just as in additive knapsacks, this problem is NP-hard in general but can be solved efficiently in some situations; the secret key enabling precisely to transform the ciphertext into an easily-solvable instance.

Unlike additive knapsacks, this multiplicative knapsack doesn't lend itself to lattice reduction attacks, which completely break many additive knapsack-based cryptosystems [1, 3, 5, 11–13].

Over the past years, several NS variants were published, these notably seek to either increase efficiency [6] or extend NS to polynomial rings [11]; to the best of our knowledge, no efficient attacks against the original NS are known.

[4] This can also be described as a modular variant of the "subset product" problem.

1.2 Security Notions

A cryptosystem is semantically secure, or equivalently IND-CPA-secure [9], if there is no adversary $\mathcal{A}$ capable of distinguishing between two ciphertexts of plaintexts of his choosing.

To capture this notion, $\mathcal{A}$ starts by creating two messages m_0 and m_1 and sends them to a challenger $\mathcal{C}$. $\mathcal{C}$ randomly selects one of the m_i (hereafter m_b) and encrypts it into a ciphertext c. $\mathcal{A}$ is then challenged with c and has to guess b with probability significantly higher than $1/2$.

Given a public-key cryptosystem $\mathsf{PKC} = \{\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt}\}$, this security notion can be formally defined by the following game:

Definition 2 (IND-CPA-Security). *The following game is played:*

- $\mathcal{C}$ *selects a secret random bit* b*;*
- $\mathcal{A}$ *outputs two messages* m_0 *and* m_1*;*
- $\mathcal{C}$ *sends to* $\mathcal{A}$ *the ciphertext* $c \leftarrow \mathsf{Encrypt}(m_b)$*;*
- $\mathcal{A}$ *outputs a guess* b'*.*

$\mathcal{A}$ *wins the game if* $b' = b$*. The advantage of* $\mathcal{A}$ *in this game is defined as:*

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKC},\mathcal{A}} := \left| \Pr\left[b = b'\right] - \frac{1}{2} \right|$$

A public-key cryptosystem PKC *is* IND-CPA-secure *if* $\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKC},\mathcal{A}}$ *is negligible for all PPT adversaries* $\mathcal{A}$*.*

IND-CPA-security is a very basic requirement, and in some scenarios it is desirable to have stronger security notions, capturing stronger adversaries. The strongest security notion for a public-key cryptosystem is indistinguishability under adaptive chosen ciphertext attacks, or IND-CCA2-security. IND-CCA2 is also defined in terms of a game, where $\mathcal{A}$ is furthermore given access to an encryption oracle and a decryption oracle:

Definition 3 (IND-CCA2-Security). *An adversary* $\mathcal{A}$ *is given access to an encryption oracle* $\mathcal{O}_E$ *and a decryption oracle* $\mathcal{O}_D$*. The following game is played:*

- $\mathcal{C}$ *selects a secret random bit* b*;*
- $\mathcal{A}$ *queries* $\mathcal{O}_E$ *and* $\mathcal{O}_D$ *and outputs two messages* m_0 *and* m_1*;*
- $\mathcal{C}$ *sends to* $\mathcal{A}$ *the ciphertext* $c \leftarrow \mathsf{Encrypt}(m_b)$*;*
- $\mathcal{A}$ *queries* $\mathcal{O}_E$ *and* $\mathcal{O}_D$ *and outputs a guess* b'*.*

$\mathcal{A}$ *wins the game if* $b' = b$ *and if no query to the oracles concerned* m_0 *nor* m_1*. The advantage of* $\mathcal{A}$ *in this game is defined as*

$$\mathsf{Adv}^{\mathsf{IND\text{-}CCA2}}_{\mathsf{PKC},\mathcal{A}} := \left| \Pr\left[b = b'\right] - \frac{1}{2} \right|$$

A public-key cryptosystem PKC *is* IND-CCA2-secure *if* $\mathsf{Adv}^{\mathsf{IND\text{-}CCA2}}_{\mathsf{PKC},\mathcal{A}}$ *is negligible for all PPT adversaries* $\mathcal{A}$*.*

We further remind the syntax of a perfectly re-randomizable encryption scheme [4, 10,16]. A perfectly re-randomizable encryption scheme consists in four polynomial-time algorithms (polynomial in the implicit security parameter k):

1. KeyGen: a randomized algorithm which outputs a public key pk and a corresponding private key sk.
2. Encrypt: a randomized encryption algorithm which takes a plaintext m (from a plaintext space) and a public key pk, and outputs a ciphertext c.
3. ReRand: a randomized algorithm which takes a ciphertext c and outputs another ciphertext c'; c' decrypts to the same message m as the original ciphertext c.
4. Decrypt: a deterministic decryption algorithm which takes a private key sk and a ciphertext c, and outputs either a plaintext m or an error indicator $\perp$.

In other words:

$$\{\mathsf{sk}, \mathsf{pk}\} \leftarrow \mathsf{KeyGen}(1^k)$$

$$\mathsf{Decrypt}(\mathsf{ReRand}(\mathsf{Encrypt}(m, \mathsf{pk}), \mathsf{pk}), \mathsf{sk}) = \mathsf{Decrypt}(\mathsf{Encrypt}(m, \mathsf{pk}), \mathsf{sk}) = m$$

Note that ReRand takes only a ciphertext and a public key as input, and in particular, does not require sk.

2 Higher-Residues Naccache-Stern

The deterministic nature of NS prevents it from achieving IND-CPA-security: Indeed, a given message m_0 will always produce the same ciphertext c_0, so $\mathcal{A}$ will always win the game of Definition 2.

We now describe an NS variant that is randomized. We then show how this modification guarantees semantic security, and even CCA2 security in the random oracle model, assuming the hardness of solving the multiplicative knapsack described earlier. In doing so, we must be very careful not to introduce additional structure that an adversary could leverage. To make this very visible, we decomposed the construction into three steps, each step pointing out the flaws avoided in the final construction.

2.1 Construction Step ①

Because the modified cryptosystem uses special prime moduli, algorithms Setup and KeyGen are merged into one single Setup + KeyGen algorithm[5].

– Setup + KeyGen: Pick a large prime p such that $(p-1)/2 = as$ is a factoring-resistant RSA modulus.

[5] Alternatively, we can regard Setup as a *pro forma* empty algorithm.

Pick a positive integer n. Let $\mathfrak{P} = \{p_0 = 2, \ldots, p_{n-1}\}$ be the set of the n first primes, so that

$$\prod_{i=0}^{n-1} p_i < p$$

Set

$$v_i = \sqrt[s]{p_i} \bmod p$$

Let g be a generator of $\mathbb{F}_p$, and $\ell = g^{2a} \bmod p$.
The public key is $(p, n, \ell, v_0, \ldots, v_{n-1})$. The private key is s.

– Encrypt: To encrypt m, pick a random integer $k \in [1, p-2]$ and compute:

$$c = \ell^k \prod_{i=0}^{n-1} v_i^{m_i} \bmod p$$

where m_i is the i-th bit of the message m.

– Decrypt: To decrypt c compute

$$m = \sum_{i=0}^{n-1} 2^i \mu_i(c, s, p).$$

To understand why decryption works we first observe that

$$(\ell^k)^s = ((g^{2a})^k)^s = g^{k(p-1)} = 1 \bmod p.$$

Hence:

$$c^s = \left(\ell^k \prod_{i=0}^{n-1} v_i^{m_i}\right)^s = \cancel{(\ell^k)^s} \prod_{i=0}^{n-1} p_i^{m_i} \bmod p.$$

And we are brought back to the original NS decryption process.

The problem: The (attentive) reader could have noted at this step that because s is large and because the p_i are very few, the odds that a p_i is an s-th residue modulo p are negligible. Hence, unless p is constructed in a very particular way, key pairs simply... cannot be constructed.[6]

A solution consisting in using a specific p and is detailed in Section 4. The alternative consists in proceeding with ② hereafter.

2.2 Construction Step ②

The workaround will be the following: Assume that we pick a v_i at random, raise it to the power s and get some integer π:

$$\pi = v_i^s \bmod p$$

[6] Note that this is obviously not be an issue with the original NS scheme.

Refresh v_i until $\pi = 0 \bmod p_i$ where π is considered as an element of $\mathbb{Z}$. (In the worst case this takes p_i trials.) Letting $y_i = \pi/p_i$, we have:

$$p_i \times y_i = v_i^s \bmod p \Rightarrow p_i = y_i^{-1} \times v_i^s = u_i \times v_i^s \bmod p$$

We will now add the u_i as auxiliary public keys.

- $\mathsf{Setup} + \mathsf{KeyGen}$: Pick a large prime p such that $(p-1)/2 = as$ is a factoring-resistant RSA modulus.

  Pick a positive integer n. Let $\mathfrak{P} = \{p_0 = 2, \ldots, p_{n-1}\}$ be the set of the n first primes, so that

  $$\prod_{i=0}^{n-1} p_i < p$$

  Generate the u_i, v_i pairs as previously described so that:

  $$p_i = u_i \times v_i^s \bmod p$$

  Let g be a generator of $\mathbb{F}_p$, and $\ell = g^{2a} \bmod p$.
  The public key is $(p, n, \ell, u_0, \ldots, u_{n-1}, v_0, \ldots, v_{n-1})$. The private key is s.

- $\mathsf{Encrypt}$: To encrypt m, pick a random integer $k \in [1, p-2]$ and compute:

  $$c_0 = \ell^k \prod_{i=0}^{n-1} v_i^{m_i} \bmod p \quad \text{and} \quad c_1 = \prod_{i=0}^{n-1} u_i^{m_i}$$

  where m_i is the i-th bit of the message m.

- $\mathsf{Decrypt}$: To decrypt c_0, c_1 compute

  $$m = \sum_{i=0}^{n-1} 2^i \eta_i(c_0, c_1, s, p)$$

Where

$$\eta_i(c_0, c_1, s, p) = \frac{\gcd(p_i, c_1 \times c_0^s \bmod p) - 1}{p_i - 1}.$$

To understand why decryption works remind that $(\ell^k)^s = 1 \bmod p$ and hence

$$c_1 \times c_0^s = \prod_{i=0}^{n-1} u_i^{m_i} \left(\ell^k \prod_{i=0}^{n-1} v_i^{m_i} \right)^s = \cancel{(\ell^k)^s} \prod_{i=0}^{n-1} (u_i v_i^s)^{m_i} = \prod_{i=0}^{n-1} p_i^{m_i} \bmod p.$$

And we are brought back to the original NS decryption process.

The problem: The (very attentive) reader could have noted that the resulting cryptosystem *does not* achieve semantic security because the construction process of c_1 is deterministic.

2.3 Construction Step ③

The workaround is the following: we provide the sender with two extra elements of $\mathbb{Z}_p$ that will allow him to blind c_0, c_1.

To that end, pick a random $\alpha \in \mathbb{Z}_p$, let $\beta\alpha^s = 1 \bmod p$ and add α, β to the public key.

The algorithms $\mathsf{Setup} + \mathsf{KeyGen}$ and $\mathsf{Decrypt}$ remain otherwise unchanged but $\mathsf{Encrypt}$ now becomes:

– $\mathsf{Encrypt}$: To encrypt m, pick a random integer $k \in [1, p-2]$ and compute:

$$c_0 = \alpha^k \prod_{i=0}^{n-1} v_i^{m_i} \bmod p \quad \text{and} \quad c_1 = \beta^k \prod_{i=0}^{n-1} u_i^{m_i}.$$

To understand why decryption works we note that (modulo p):

$$c_1 \times c_0^s = \beta^k \prod_{i=0}^{n-1} u_i^{m_i} \left(\alpha^k \prod_{i=0}^{n-1} v_i^{m_i} \right)^s = \cancel{(\beta\alpha^{s})^k} \prod_{i=0}^{n-1} (u_i v_i^s)^{m_i} = \prod_{i=0}^{n-1} p_i^{m_i}.$$

And we are brought back to the original NS decryption process.

3 Security

3.1 Semantic Security

The modified scheme's security essentially relies on blinding an NS ciphertext using a multiplicative factor $\ell^k = g^{2ka} \bmod p$, which belongs to the subgroup of $\mathbb{Z}_p$ of order b.

Lemma 1. *Under the subgroup hiding assumption in $\mathbb{Z}_p$, the scheme described in Section 2.1 is* $\mathsf{IND\text{-}CPA}$*-secure.*

Recall that the subgroup-hiding assumption [2] states that the uniform distribution over $\mathbb{Z}_p$ is indistinguishable from the uniform distribution over one of its subgroups.

Proof. Assume that $\mathcal{A}(\mathsf{pk})$ wins the $\mathsf{IND\text{-}CPA}$ game with non-negligible advantage. Then in particular $\mathcal{A}(\mathsf{pk})$ has non-negligible advantage in the "real-or-random" game

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{R/R}} := \Pr[\mathcal{A}^{\mathcal{E}_{\mathsf{pk}}}(\mathsf{pk}) = 1] - \Pr[\mathcal{A}^{\mathcal{O}}(\mathsf{pk}) = 1]$$

where $\mathcal{E}_{\mathsf{pk}}$ is an encryption oracle and $\mathcal{O}$ is a random oracle. We define $\mathcal{B}(\mathsf{pk}, \gamma)$ as follows:

– Let $\mathcal{E}_{\mathcal{B}}(m) = \gamma \prod_{i=0}^{n-1} v_i^{m_i} \bmod p$;
– $\mathcal{B}(\mathsf{pk}, \gamma)$ returns the same result as $\mathcal{A}^{\mathcal{E}_{\mathcal{B}}}(\mathsf{pk})$

The scenario $\mathcal{B}(\mathsf{pk}, \gamma = g^{2au})$ yields $\mathcal{E}_{\mathcal{B}} = \mathcal{E}_{\mathsf{pk}}$. The scenario $\mathcal{B}(\mathsf{pk}, \gamma = g^{u})$ for random u gives a ciphertext that is a uniform value, and therefore behaves as a perfect simulator of a random oracle, i.e. $\mathcal{E}_{\mathcal{B}} = \mathcal{O}$. Hence if $\mathcal{A}$ is an efficient adversary against our scheme, then $\mathcal{B}$ is an efficient solver for the subgroup-hiding problem. □

Note that this part of the argument does not fundamentally rely on the original NS being secure — indeed, we may consider an encryption scheme that produces ciphertexts of the form $c = x^k m$. Decryption for such a cryptosystem would be tricky, as $c^b = m^b$ and there are b possible roots. That is why using NS is useful, as we do not have decryption ambiguity issues.

As we pointed out, the construction of Section 2.2 is not semantically secure: indeed, c_1 is generated deterministically from m. This is addressed in Section 2.3 by introducing two numbers α and β. Using a similar argument as in Lemma 1, we have

Lemma 2. *Under the DDH assumption in $\mathbb{Z}_p$, and assuming that factoring $(p-1)/2$ is infeasible, the scheme described in Section 2.3 is* IND-CPA*-secure.*

Note that these hypotheses can be simultaneously satisfied.

3.2 CCA2 Security

Even more interesting is the case for security against adaptive chosen-ciphertext attacks (IND-CCA2) [7, 8].

The original NS is naturally not IND-CCA2; nor is in fact the "Step ①" variant discussed above: indeed it is possible to re-randomise a ciphertext, which immediately gives a way to win the IND-CCA2 game.

To remedy this, we leverage the fact that upon successful decryption, we can recover the randomness ℓ^k. The idea is to choose k in some way that depends on m_i. If k is a deterministic function of m_i only however, randomisation is lost. Therefore we suggest the following variant, at the cost of some bandwidth:

- Instead of m, we encrypt a message $m\|r$ where r is a random string.
- Let $k \leftarrow H(m\|r)$ where H is a cryptographic hash function, and use this value of k instead of choosing it randomly in Encrypt.
- Modify Decrypt to recover ℓ^k (or α^k and β^k). Upon successfully recovering $(m\|r)$, extract r, and check that ℓ^k (resp. α^k and β^k) correspond to the correct value of k — otherwise it ouputs $\perp$.

This approach guarantees IND-CCA2 in the random oracle model; this can be captured as a series of games:

- *Game 0:* This is the IND-CCA2 game against our scheme (① or ③), instantiated with some hash function H.
- *Game 1:* This game differs from *Game 0* in replacing H by a random oracle $\mathcal{O}$. In the random oracle model, this game is computationally indistinguishable from *Game 0*.

– *Game 2:* This game differs from *Game 1* by the fact that the ciphertext is replaced by an uniformly-sampled random element of the ciphertext space. The results on IND-CPA security tell us that this game is computationally indistinguishable from *Game 1* (under their respective hypotheses).

4 Generating Strong Pseudo-Primes in Several Bases

We now backtrack and turn our attention to generating specific moduli allowing to implement securely the "①" scheme of Section 2.1. This boils down to describing how to efficiently generate strong pseudo-prime numbers. In this section, we denote N the sought-after modulus.

Using quadratic reciprocity, we first introduce an algorithm generating numbers passing Fermat's test. Then we leverage quartic reciprocity to generate numbers passing Miller-Rabin's test. The pseudoprimes we need must be strong over several bases, and complexity is polynomial in the size of the product of these bases.

4.1 Primality Tests

A base-A Fermat primality test consists in checking that $A^B \equiv A \bmod B$. Every prime passes this test for all bases A. There are however composite numbers, known as Carmichael numbers, that also pass this test in all bases. For instance, $1729 = 7 \cdot 13 \cdot 19$ is such a number. There are an infinity of Carmichael numbers.

The Miller-Rabin primality test also relies on Fermat's little theorem. Let $B - 1 = 2^e m$ with m odd. An integer B passes the Miller-Rabin test if $A^m \equiv 1 \bmod B$ or if there exists an $i \leq e - 1$ such that $A^{2^i m} \equiv -1 \bmod B$.

Definition 4 (Strong pseudo-prime). *A number that passes the Miller-Rabin test is said strongly pseudo-prime in base A.*

An interesting theorem [14, Proposition 2] [17] states that a composite number can only be strongly pseudo-prime for a quarter of the possible bases.

4.2 Constructing Pseudo-Primes

When p and $2p - 1$ are prime, Fermat's test amounts to the computing of a Jacobi symbol. Indeed,

Theorem 1. *Let p be a prime such that $q = 2p - 1$ is also prime. Let $A \in \mathrm{QR}_q$. Then $B = pq$ passes Fermat's test in base A.*

Proof.

$$A^B \equiv (A^p)^q \equiv A^q \equiv A^{2(p-1)+1} \equiv A \bmod p$$

$$A^B \equiv (A^q)^p \equiv A^p \equiv A^{(q-1)/2+1} \equiv A \left(\frac{A}{q}\right) \equiv A \bmod q$$

By the Chinese remainder theorem, we find that $A^B \equiv A \bmod B$. □

From Gauss' quadratic reciprocity theorem, if $q \equiv 1 \bmod 4$ we can take $q \equiv 1 \bmod A$ which guarantees that $A \in \mathrm{QR}_q$. To make 2 a quadratic residue modulo q we must have $q \equiv \pm 1 \bmod 8$. It is therefore easy to construct numbers that pass Fermat's test in a prescribed list of bases.

4.3 Constructing Strong Pseudo-primes

In this section we seek to generate numbers that are strongly pseudo-prime in base η, where η is prime. Let p denote a prime number such that $q = 2p - 1$ is also prime, and $N = pq$. We have the following equations:

$$\begin{aligned} N - 1 &\equiv 0 \bmod p - 1 \\ N - 1 &\equiv \frac{q-1}{2} \bmod q - 1 \\ \frac{n-1}{2} &\equiv \frac{p-1}{2} \bmod p - 1 \\ \frac{n-1}{2} &\equiv 3\frac{q-1}{4} \bmod q - 1 \end{aligned}$$

From there on, we will use the notation $\left(\frac{\cdot}{\cdot}\right)_4$ to denote the quartic residue symbol.

Theorem 2. *Let p be a prime such that $q = 2p - 1 \equiv 1 \bmod 8$ is also prime. Let A be an integer such that*

$$\left(\frac{A}{p}\right) = -1, \qquad \left(\frac{A}{q}\right) = +1, \quad \textit{and} \quad \left(\frac{A}{q}\right)_4 = -1.$$

Then $N = pq$ passes the Miller-Rabin test in base A.

Proof. Note that if $A^{(N-1)/2} \equiv -1 \bmod N$, then n passes the Miller-Rabin test in base a. It then suffices to compute this quantity modulo p and q respectively:

$$\begin{aligned} A^{(N-1)/2} &\equiv A^{(p-1)/2} \equiv \left(\frac{A}{p}\right) \equiv -1 \bmod p \\ A^{(N-1)/2} &\equiv A^{3(q-1)/4} \equiv \left(\frac{A}{p}\right)_4^3 \equiv -1 \bmod q. \end{aligned}$$

□

Bases $\eta > 5$. Let $\eta \geq 7$ be a prime number. We consider here the case $p \equiv 5 \bmod 8$, i.e. $q \equiv 9 \bmod 16$. We will leverage the following classical result:

Theorem 3. *Let q be a prime number, $q = A^2 + B^2 \equiv 1 \bmod 8$ with B even. Let η be a prime number such that $(p/\eta) = 1$, then*

$$\left(\frac{\eta}{q}\right)_4 = 1 \Leftrightarrow \begin{cases} \eta \mid B & \text{, or} \\ \eta \mid A \textit{ and } \left(\frac{2}{\eta}\right) = 1 & \text{, or} \\ A \equiv \mu B \textit{ where } \mu^2 + 1 \equiv \lambda^2 \bmod \eta \textit{ and } \left(\frac{\lambda(\lambda+1)}{\eta}\right) = 1 \end{cases}$$

We will also need the following easy lemmata:

Lemma 3. *Let $\eta \geq 7$ be a prime number, there is at least an integer Λ such that*

$$\left(\frac{\Lambda}{\eta}\right) = \left(\frac{2-\Lambda}{\eta}\right) = -1.$$

Proof. Let

$$\begin{aligned}
s_1 &= \#\left\{i \in \mathbb{F}_\eta, \left(\frac{i}{\eta}\right) = +1, \left(\frac{2-i}{\eta}\right) = +1, \right\} \\
s_2 &= \#\left\{i \in \mathbb{F}_\eta, \left(\frac{i}{\eta}\right) = +1, \left(\frac{2-i}{\eta}\right) = -1, \right\} \\
s_3 &= \#\left\{i \in \mathbb{F}_\eta, \left(\frac{i}{\eta}\right) = -1, \left(\frac{2-i}{\eta}\right) = +1, \right\} \\
s_4 &= \#\left\{i \in \mathbb{F}_\eta, \left(\frac{i}{\eta}\right) = -1, \left(\frac{2-i}{\eta}\right) = -1, \right\}.
\end{aligned}$$

Then it is clear that $s_1 + s_2 + s_3 + s_4 = \eta - 2$. The quantity $s_1 + s_2$ corresponds to the number of quadratic residues modulo η, except maybe 2. Therefore,

$$s_1 + s_2 = \frac{\eta - \left(\frac{2}{\eta}\right)}{2} - 1.$$

By symmetry between i and $2 - i$, we have $s_2 = s_3$. We also have

$$\begin{aligned}
s_2 + s_3 &= \#\left\{i \in \mathbb{F}_\eta, \left(\frac{i(2-i)}{\eta}\right) = -1\right\} \\
&= \#\left\{i \in \mathbb{F}_\eta^*, \left(\frac{2/i - 1}{\eta}\right) = -1\right\} \\
&= \#\left\{u \in \mathbb{F}_\eta, u \neq -1, \left(\frac{u}{\eta}\right) = -1\right\} \\
&= \frac{\eta + \left(\frac{-1}{\eta}\right)}{2} - 1.
\end{aligned}$$

From that we get the value of s_4:

$$s_4 = \frac{\eta + 2\left(\frac{2}{\eta}\right) - \left(\frac{-1}{\eta}\right) - 2}{4}.$$

Therefore, for every $\eta \geq 7$, $s_4 > 0$. □

Choosing such an i, we denote λ the integer such that $i = 1 + 1/\lambda \bmod \eta$. Then,

$$\left(\frac{1+1/\lambda}{\eta}\right) = \left(\frac{1-1/\lambda}{\eta}\right) = -1$$
$$\left(\frac{(\lambda+1)\lambda}{\eta}\right) = \left(\frac{(\lambda-1)\lambda}{\eta}\right) = -1$$
$$\left(\frac{\lambda^2-1}{\eta}\right) = \left(\frac{(\lambda+1)\lambda}{\eta}\right)\left(\frac{(\lambda-1)\lambda}{\eta}\right) = 1.$$

Let μ be such that $\mu^2 + 1 = \lambda^2$. We can thus construct λ and μ so that the third possibility of Theorem 3 is never satisfied.

Lemma 4. *Let $\eta \geq 7$ be a prime number, there is at least an integer x such that $(x/\eta) = -1$ and $(2x - 1/\eta) = +1$.*

Proof. As for the previous lemma, we show that there are $\frac{1}{4}\left(\eta + 2\left(\frac{2}{\eta}\right) - \left(\frac{-1}{\eta}\right) - 2\right)$ such values of x, which strictly positive for $\eta \geq 7$. □

For such an x, we write $y = 2x - 1 = z^2 \bmod \eta$, $A_\eta = z/\lambda \bmod \eta$, and $B_\eta = A_\eta \mu$. We then have

$$A_\eta^2 + B_\eta^2 = (1 + \mu^2)A_\eta^2 = \lambda^2 A_\eta^2 = z^2 = y \bmod \eta$$

If $q = A^2 + B^2 \equiv 1 \bmod 8$ is prime, with B even, $A \equiv A_\eta \bmod \eta$, and $B \equiv B_\eta \bmod \eta$, then we see that the conditions of Theorem 3 are not satisfied, hence $(\eta/q)_4 = -1$. Furthermore, $q \equiv y \bmod \eta$ so that $(\eta/q) = +1$. If we assume that $p = (q+1)/2$ is prime, and that $p \equiv 5 \bmod 8$, then the conditions of Theorem 2 are satisfied. Indeed, $p \equiv x \bmod \eta$ so that $(\eta/p) = (x/\eta) = -1$. Thus we generated a pseudo-prime in base η.

All in all, the results from this section are captured by the following theorem.

Theorem 4. *Let $\eta \geq 7$ be a prime number. There are integers A_η, A_η such that $N = pq$ is strongly pseudo-prime in base η, provided that*

$$\begin{cases} q & = A^2 + B^2 \\ B & \textit{is even} \\ A & \equiv A_\eta \bmod \eta \\ B & \equiv B_\eta \bmod \eta \\ q & \equiv 9 \bmod 16 \\ p & = (q-1)/2 \\ q & \textit{is prime} \\ p & \textit{is prime} \end{cases}$$

Base $\eta = 2$. In that case the following theorem applies.

Theorem 5. *The integer $N = pq$ is strongly pseudo-prime in base 2 provided that*

$$\begin{cases} q & = A^2 + B^2 \\ A & \equiv 3 \bmod 8 \\ B & \equiv 4 \bmod 8 \\ p & = (q-1)/2 \\ q & \textit{is prime} \\ p & \textit{is prime} \end{cases}$$

Proof. From the conditions of Theorem 5, $q \equiv 9 \bmod 16$ and $q \equiv 5 \bmod 8$, which proves that 2 is a square modulo q and not modulo p, as it is not of the form $\alpha^2 + 64\beta^2$. □

Bases $\eta = 3$ and $\eta = 5$. In both cases, we cannot find p and q such that the base is a square modulo q and not modulo p. As we will see in the next section this is not too much of a problem in practice. We can in any case ensure that the base is a quartic residue modulo q, using for instance the following choices:

$$\begin{aligned} A_3 &= 1, & B_3 &= 0, \\ A_5 &= 1, & B_5 &= 0. \end{aligned}$$

4.4 Combining Bases

Consider a set $\mathfrak{P}$ of prime numbers, which will be used as bases. For each $\eta \in \mathfrak{P}$, we construct a_η, b_η as described in the previous section, using either the general construction (for $\eta \geq 7$) or the specific constructions (for $\eta = 2, 3, 5$). Then we invoke the Chinese remainder theorem, to get three integers $a_{\mathfrak{P}}$, $b_{\mathfrak{P}}$, and $m_{\mathfrak{P}}$ such that $N = pq$ is strongly pseudo-prime in all bases of $\mathfrak{P}$ (except maybe 3 and 5), provided that

$$\begin{cases} q & = A^2 + B^2 \\ B & \text{is even} \\ A & \equiv A_{\mathfrak{P}} \bmod m_{\mathfrak{P}} \\ B & \equiv B_{\mathfrak{P}} \bmod m_{\mathfrak{P}} \\ q & \equiv 9 \bmod 16 \\ p & = (q-1)/2 \\ q & \text{is prime} \\ p & \text{is prime} \end{cases}$$

In fact, running the algorithm several times eventually yields an integer N that is also strongly pseudo-prime in bases 3 and 5.

4.5 Numerical Example

Consider $\mathfrak{P} = \{p_1 = 2, \ldots, p_{46}\}$ the set of all primes smaller than 200. We get:

$$\begin{aligned}
A_{\mathfrak{P}} &= 2409510466413366836102939894877209385943700042913129394126042848260031865186440501 1\\
B_{\mathfrak{P}} &= 2450013656206455126042788019975083012281289375458232594038192481071092303905088660\\
m_{\mathfrak{P}} &= 3119968816673384621299679642531925550675198715961420378037212989947404614465880324 0
\end{aligned}$$

From these we get the following number N, which is strongly pseudo-prime over all the bases in $\mathfrak{P}$:

$$\begin{aligned}
p = {} & 2916185066639798364850755523754254 1271029\\
& 3572761949403490589938128447683393074939381276465948218170090252412901503716427685977614433185846920398877075011893352376431218094218664172215622 1\\
q = {} & 5832370133279596729701511047508506825420587145523898806981179876256895366786149878762559318964363401805048258030074328553719522886637169384079775415002378670475286243 6188437328344431244 1\\
N = {} & 1700827068578593046013465420408804918699647861382732351483602640070116599271370938094251080691735799378797733582218499445066465988878583613584031972656406509828930523285603156508822841342069665836703886702058844741799083951362563103117204854024938903124158459685637812694900928898660385791837913950199481739941509599211056150786127399995262142244846207324478665807217335845461
\end{aligned}$$

This N can hence be used as the missing modulus needed to instantiate a "Step ①" NS variant.

5 Extensions

5.1 Using Composite Moduli

In the ②/③ variants of our scheme, one might be tempted to replace p *itself* by an RSA modulus n, where $\phi(n) = 2ab$. Indeed, the original NS construction allows for such a choice.

Doing so, however, would immediately leak information about the factorisation of n: Indeed, $\gcd(g^a - 1, n) = p$.

There is a workaround: First we choose p and q so that $(p-1)/2$ and $(q-1)/2$ are RSA moduli, i.e. $p - 1 = 2s_1s_2$ and $q - 1 = 2r_1r_2$, with large s_1, s_2, r_1, r_2. Then we set $n = pq$, $a = s_1r_1$, and $b = 2s_2r_2$. Therefore $\phi(n) = 2ab$ as before, but the GCD attack mentioned above does not apply, and the modified ②/③ Naccache-Stern cryptosystem works.

5.2 Bandwidth Improvements

The idea described in this paper is fully compatible with the modifications introduced in [6] to improve encryption bandwidth.

But there is even more: An interesting observation is that, upon decryption, it is possible to recover both the message m and the whitening x^k. This is unlike most randomized encryption schemes, where the random nonce is lost. Thus we may contemplate storing some information in k, thereby augmenting somewhat the total information contained in a ciphertext. Alternatively, x^k may also be used as key material if NS is used (in a hybrid mode) as a key transfer mechanism.

For instance, given a message $m = m_1 \| m_2$, we may encrypt $m_1 \| k$ using the blinding m_2^k with odd k. Upon decryption, one recovers k, and computes the k-th root of the blinding factor m_2^k — such a root is unique with overwhelming probability — thereby reconstructing the whole message.

One nontrivial research direction is to provide, in the message m, *hints* that make solving the discrete log modulo p easier and thereby embed directly information in k.

References

1. Adleman, L.M.: On breaking the iterated Merkle-Hellman public-key cryptosystem. In: Chaum, D., Rivest, R.L., Sherman, A.T. (eds.) Advances in Cryptology – CRYPTO'82. pp. 303–308. Plenum Press, New York, USA, Santa Barbara, CA, USA (1982)
2. Boneh, D., Goh, E.J., Nissim, K.: Evaluating 2-DNF formulas on ciphertexts. In: Kilian, J. (ed.) TCC 2005: 2nd Theory of Cryptography Conference. Lecture Notes in Computer Science, vol. 3378, pp. 325–341. Springer, Heidelberg, Germany, Cambridge, MA, USA (Feb 10–12, 2005)
3. Brickell, E.F.: Breaking iterated knapsacks. In: Blakley, G.R., Chaum, D. (eds.) Advances in Cryptology – CRYPTO'84. Lecture Notes in Computer Science, vol. 196, pp. 342–358. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 19–23, 1984)

4. Canetti, R., Krawczyk, H., Nielsen, J.B.: Relaxing chosen-ciphertext security. In: Boneh, D. (ed.) Advances in Cryptology – CRYPTO 2003. Lecture Notes in Computer Science, vol. 2729, pp. 565–582. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 17–21, 2003)
5. Chee, Y.M., Joux, A., Stern, J.: The cryptoanalysis of a new public-key cryptosystem based on modular Knapsacks. In: Feigenbaum, J. (ed.) Advances in Cryptology – CRYPTO'91. Lecture Notes in Computer Science, vol. 576, pp. 204–212. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 11–15, 1992)
6. Chevallier-Mames, B., Naccache, D., Stern, J.: Linear bandwidth Naccache-Stern encryption. In: Ostrovsky, R., Prisco, R.D., Visconti, I. (eds.) SCN 08: 6th International Conference on Security in Communication Networks. Lecture Notes in Computer Science, vol. 5229, pp. 327–339. Springer, Heidelberg, Germany, Amalfi, Italy (Sep 10–12, 2008)
7. Cramer, R., Shoup, V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Krawczyk, H. (ed.) Advances in Cryptology – CRYPTO'98. Lecture Notes in Computer Science, vol. 1462, pp. 13–25. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 23–27, 1998)
8. Fujisaki, E., Okamoto, T., Pointcheval, D., Stern, J.: RSA-OAEP is secure under the RSA assumption. Journal of Cryptology 17(2), 81–104 (Mar 2004)
9. Goldwasser, S., Micali, S.: Probabilistic encryption and how to play mental poker keeping secret all partial information. In: Lewis, H.R., Simons, B.B., Burkhard, W.A., Landweber, L.H. (eds.) Proceedings of the 14th Annual ACM Symposium on Theory of Computing, May 5-7, 1982, San Francisco, California, USA. pp. 365–377. ACM (1982)
10. Groth, J.: Rerandomizable and replayable adaptive chosen ciphertext attack secure cryptosystems. In: Naor, M. (ed.) TCC 2004: 1st Theory of Cryptography Conference. Lecture Notes in Computer Science, vol. 2951, pp. 152–170. Springer, Heidelberg, Germany, Cambridge, MA, USA (Feb 19–21, 2004)
11. Herold, G., Meurer, A.: New attacks for knapsack based cryptosystems. In: Visconti, I., Prisco, R.D. (eds.) SCN 12: 8th International Conference on Security in Communication Networks. Lecture Notes in Computer Science, vol. 7485, pp. 326–342. Springer, Heidelberg, Germany, Amalfi, Italy (Sep 5–7, 2012)
12. Joux, A., Stern, J.: Cryptanalysis of another knapsack cryptosystem. In: Imai, H., Rivest, R.L., Matsumoto, T. (eds.) Advances in Cryptology – ASIACRYPT'91. Lecture Notes in Computer Science, vol. 739, pp. 470–476. Springer, Heidelberg, Germany, Fujiyoshida, Japan (Nov 11–14, 1993)
13. Lenstra Jr., H.W.: On the Chor-Rivest knapsack cryptosystem. Journal of Cryptology 3(3), 149–155 (1991)
14. Monier, L.: Evaluation and comparison of two efficient probabilistic primality testing algorithms. Theoretical Computer Science 12(1), 97–108 (1980)
15. Naccache, D., Stern, J.: A new public-key cryptosystem. In: Fumy, W. (ed.) Advances in Cryptology – EUROCRYPT'97. Lecture Notes in Computer Science, vol. 1233, pp. 27–36. Springer, Heidelberg, Germany, Konstanz, Germany (May 11–15, 1997)
16. Prabhakaran, M., Rosulek, M.: Rerandomizable RCCA encryption. In: Menezes, A. (ed.) Advances in Cryptology – CRYPTO 2007. Lecture Notes in Computer Science, vol. 4622, pp. 517–534. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 19–23, 2007)
17. Rabin, M.O.: Probabilistic algorithm for testing primality. Journal of number theory 12(1), 128–138 (1980)