On-Line/Off-Line DCR-based Homomorphic Encryption and Applications

Marc Joye[0000−0003−4433−2333]

Zama, Paris, France
marc@zama.ai

Abstract. On-line/off-line encryption schemes enable the fast encryption of a message from a pre-computed coupon. The paradigm was put forward in the case of digital signatures.
This work introduces a compact public-key additively homomorphic encryption scheme. The scheme is semantically secure under the decisional composite residuosity (DCR) assumption. Compared to Paillier cryptosystem, it merely requires one or two integer additions in the on-line phase and no increase in the ciphertext size. This work also introduces a compact on-line/off-line trapdoor commitment scheme featuring the same fast on-line phase. Finally, applications to chameleon signatures are presented.

Keywords: On-line/off-line encryption · Additively homomorphic encryption · Decisional composite residuosity assumption · Trapdoor commitments · Chameleon signatures

1 Introduction

A number of applications can afford slower computations as long as they are not required to be performed on-line. Most public-key encryption schemes entail the evaluation of many modular multiplications with a large modulus as part of the encryption procedure. Furthermore, certain applications like electronic voting or private data analytics require operating on ciphertexts. Additively homomorphic encryption enables to directly adding ciphertexts or, more generally, evaluating linear combinations thereof. This is in contrast with traditional encryption schemes where data first needs to be decrypted prior to being processed.

On-line/off-line encryption Real-time encryption necessitates the encryption process to be as fast as possible. This leads to the notion of *on-line/off-line cryptography* introduced by Even *et al.* [10] for digital signatures.

In an on-line/off-line encryption scheme, the encryption process is divided into two phases. The first phase, performed *off-line*, is independent of the message to be encrypted. Examples include a server pre-computing values at idle time or a low-end hardware token with pre-computed values stored in memory. The second phase, performed *on-line*, takes on input a value pre-computed in the off-line phase and a message and produces a ciphertext. Only the on-line phase is required to be fast.

Paillier's additive encryption The *Paillier cryptosystem* [17] is a public-key encryption scheme. The public key is RSA-type modulus $N = pq$ where p and q are two large secret equal-size primes. The message space $\mathcal{M}$ is the additive group $\mathbb{Z}/N\mathbb{Z}$. The encryption of a plaintext message $m \in \mathcal{M}$ is given by $C = (1+N)^m \, r^N \bmod N^2$ for some uniformly random integer $r \xleftarrow{\$} [1, N)$ with $\gcd(r, N) = 1$.[1] Paillier cryptosystem is known to provide indistinguishability of encryptions (semantic security) under the DCR assumption.

A salient feature of the system resides its additive property: given the encryption of two plaintext messages m_1 and m_2, there is an efficient public procedure providing an encryption of $m_1 + m_2$ (as an element of the message space). Specifically, letting $C_1 = (1+N)^{m_1} \, {r_1}^N \bmod N^2$ and $C_2 = (1+N)^{m_2} \, {r_2}^N \bmod N^2$, it turns out that $C_3 \coloneqq C_1 \, C_2 \, \rho^N \bmod N^2$ for any $\rho \in [1, N)$ with $\gcd(\rho, N) = 1$ is an encryption of $m_1 + m_2 \pmod N$.

Trapdoor commitments Commitment schemes can be derived from semantically secure encryption schemes [11]. A *commitment scheme* is a cryptographic primitive allowing a user to commit to a chosen value m, with the ability to reveal the committed value later. The resulting commitment C to m must be such that it hides the value of m. Further, it should not be possible for the user to exhibit a value $m' \neq m$ that results in the same commitment C.

As the name suggests, a *trapdoor commitment scheme* [5] is a commitment scheme with some secret trapdoor. The knowledge of the trapdoor enables opening a commitment C to any *chosen* value m'. This feature is known as the "chameleon" property, a term coined in [5]. Non-interactive trapdoor commitment schemes naturally give rise to *chameleon hash functions* [14,3]. Chameleon hash functions are hash functions associated with a pair of hashing/trapdoor keys. Again, the name chameleon refers to the ability for the owner of the trapdoor key to modify the input without changing the output. A useful application of chameleon hashing is *chameleon signatures* [14].

Our contributions Additive encryption schemes can easily be turned into efficient on-line/off-line encryption schemes. Specifically, for the Paillier cryptosystem, a message $m \in \mathbb{Z}/N\mathbb{Z}$ can be encrypted using an hybrid approach, where the ciphertext is set as the pair (C_1, c_2) with

$$C_1 = (1+N)^{\mu} \, r^N \bmod N^2 \quad \text{and} \quad c_2 = (m + \mu) \bmod N$$

for some random mask $\mu \in \mathbb{Z}/N\mathbb{Z}$. Ciphertext C_1 is a regular Paillier encryption that can be pre-computed ahead of time. From (C_1, μ), the on-line phase only involves a modular addition to get the second component c_2 of the ciphertext. On the downside, the resulting ciphertext (C_1, c_2) is longer than a regular Paillier ciphertext: $3 \log_2 N$ bits instead of $2 \log_2 N$ bits.

This paper presents equally efficient Paillier-like constructions but without increasing the ciphertext size. In order to do so, we introduce a new operator

[1] In practice, there is no need to check that $\gcd(r, N) = 1$. This condition is verified with overwhelming probability, namely with probability $1 - \frac{1}{N-1-\#(\mathbb{Z}/N\mathbb{Z})^*} > 1 - \frac{1}{\sqrt{N}}$.

that we call the "Ups" function as its relates, modulo some fixed integer N, the upper part of an integer to the integer itself. This operator is the heart of our constructions.

In the Paillier cryptosystem, if $(1+N)^m \bmod N^2$ is evaluated as $1+m\,N$, we see that the encryption of message m can be obtained as

$$C = (1+m\,N)\,R \bmod N^2$$

where $R = r^N \bmod N^2$. Hence, if the value of R is pre-computed, producing a Paillier's ciphertext C essentially costs an integer multiplication plus a multiplication modulo N^2. It is useful to note that R is a Paillier encryption of 0 and that $(1+m\,N)$ is a trivial Paillier encryption of m (i.e., using $r=1$). Ciphertext C can therefore be seen as the homomorphic addition of plaintexts 0 and m. Using the Ups function $\varUpsilon_N$, a ciphertext is expressed as a pair of two integers modulo N. In particular, letting $[R]_N = R \bmod N$, R is represented as $([R]_N, \varUpsilon_N(R)) \in (\mathbb{Z}/N\mathbb{Z})^* \times \mathbb{Z}/N\mathbb{Z}$ and $(1+m\,N)$ as $(1,m)$. Interestingly, their homomorphic addition, $([R]_N, \varUpsilon_N(R)) \boxplus (1,m)$, leads to the pair $([R]_N, \varUpsilon_N(R)+m) \in (\mathbb{Z}/N\mathbb{Z})^* \times \mathbb{Z}/N\mathbb{Z}$, which represents ciphertext C. We exploit this property of the Ups function to design an efficient on-line/off-line homomorphic encryption scheme. The off-line comprises the pre-computation of coupons of the form $([R]_N, \varUpsilon_N(R))$ while the on-line phase just add $m \pmod N$ to the second component of a fresh coupon to get an encryption of a plaintext message m.

The on-line/off-line encryption scheme we propose is semantically secure. In addition, its encryption function induces a trapdoor permutation on $(\mathbb{Z}/N\mathbb{Z})^* \times \mathbb{Z}/N\mathbb{Z}$ given by

$$\begin{aligned}\pi\colon (\mathbb{Z}/N\mathbb{Z})^* \times \mathbb{Z}/N\mathbb{Z} &\longrightarrow (\mathbb{Z}/N\mathbb{Z})^* \times \mathbb{Z}/N\mathbb{Z},\\ (r,m) &\longmapsto (u,v) = \bigl(r^N \bmod N, m + \varUpsilon_N(r^N \bmod N^2)\bigr)\end{aligned}$$

and comes with an efficiently computable inverse map $\pi^{-1}\colon (\mathbb{Z}/N\mathbb{Z})^* \times \mathbb{Z}/N\mathbb{Z} \to (\mathbb{Z}/N\mathbb{Z})^* \times \mathbb{Z}/N\mathbb{Z}, (u,v) \mapsto (r,m)$ with $r = u^{1/N} \bmod N$ and $m = v - \varUpsilon_N(r^N \bmod N^2)$. Abstracting the scheme given in [7, § 6.1], we show how the presence of such a map π^{-1} in an homomorphic encryption scheme allows one to get a trapdoor commitment scheme. We adapt this result to our homomorphic encryption scheme and so obtain a concrete instantiation of an efficient on-line/off-line trapdoor commitment scheme. The resulting scheme inherits the fast on-line phase of the encryption scheme. Non-interactive versions of the commitment scheme are applied to design chameleon signatures that are free of key exposure [1,9,2].

Outline of the paper The rest of the paper is organized as follows. The next section defines the Ups function and studies its arithmetic properties. Section 3 presents an efficient on-line/off-line encryption scheme. It also details its homomorphic operations. The security proofs are deferred to Appendix B. A companion on-line/off-line trapdoor commitment scheme is proposed in Section 4. It is applied as a building block for secure chameleon signatures. Finally, Section 5 concludes the paper.

2 The "Ups" Function

Throughout this section, we fix a positive integer N.

For a real number r, the floor function $\lfloor r \rfloor$ returns the greatest integer less than or equal to r. For example, $\lfloor 3.1415 \rfloor = 3$ and $\lfloor -3.1415 \rfloor = -4$. For an integer x, $\lfloor x/N \rfloor$ denotes the integer division of x by N and $x \bmod N$ denotes the remainder of the division of x by N. Clearly, $x \bmod N = x - N \cdot \lfloor x/N \rfloor \in [0, N)$. For a rational number $\frac{a}{b}$ with $a, b \in \mathbb{Z}$ and $\gcd(b, N) = 1$, $\frac{a}{b} \bmod N = a \cdot b^{\star} \bmod N$ where $b^{\star}$ is the inverse of b modulo N; i.e., $b^{\star}$ is an integer satisfying $b \cdot b^{\star} \equiv 1 \pmod{N}$. The integer $b^{\star} = b^{-1} \bmod N$ can be obtained via the extended Euclidean algorithm (see e.g. [16, Algorithm 2.107]).

Definition 1. *The* Ups function w.r.t. N, *denoted by* Υ_N, *takes as input an integer that is co-prime to* N *and returns a value in* $\mathbb{Z}/N\mathbb{Z}$; *it is given by*

$$\Upsilon_N \colon x \mapsto \Upsilon_N(x) = \frac{\lfloor x/N \rfloor}{x} \bmod N \enspace .$$

Let $x \in \mathbb{Z}$ with $\gcd(x, N) = 1$. The Ups function satisfies the following properties:

1. $\Upsilon_N(x) = \Upsilon_N(x \bmod N^2) = \frac{\frac{x \bmod N^2}{x \bmod N} - 1 \mod N^2}{N}$;
2. $\Upsilon_N(-1) = 1$;
3. $\Upsilon_N(x) = 0$ if $x \bmod N^2 < N$; in particular, $\Upsilon_N(1) = 0$;
4. $\Upsilon_N(-x) = \Upsilon_N(x) + x^{-1} \mod N$.

Proof. 1. The first property is a consequence of $\lfloor x/N \rfloor \equiv \lfloor (x \bmod N^2)/N \rfloor \bmod N$ and $x \equiv (x \bmod N^2) \pmod{N}$. We so have $\Upsilon_N(x) = \Upsilon_N(x \bmod N^2)$. Now write $x \bmod N^2 = x_l + x_h N$ with $0 \le x_l, x_h < N$. Clearly, we have $x \bmod N^2 \equiv x_l + x_h N \equiv x_l[1 + (x_h \cdot {x_l}^{-1} \bmod N)N] \equiv x_l[1 + \Upsilon_N(x)N] \pmod{N^2}$ and so $\Upsilon_N(x) = (x \cdot {x_l}^{-1} - 1 \mod N^2)/N$.
2. The second property follows from $\lfloor -1/N \rfloor = -1$. Hence, $\Upsilon_N(-1) = \frac{-1}{-1} \bmod N = 1$.
3. If $x \bmod N^2 < N$ then $x \bmod N^2 = x \bmod N$. From the first property, we then have $\Upsilon_N(x) = \frac{0 \bmod N^2}{N} = 0$.
4. Multiplying through by x, the last property boils down to $x \cdot \Upsilon_N(-x) \equiv x \cdot \Upsilon_N(x) + 1 \pmod{N}$, which immediately follows from $-\lfloor -x/N \rfloor = \lfloor x/N \rfloor + 1$.

Remark 1. As alluded in the above proof, a positive integer $x < N^2$, co-prime to N, can uniquely be put under the form $x = x_l + x_h N$ with $x_l = x \bmod N$ and $x_h = \lfloor x/N \rfloor$. For such an integer, the Ups function can equivalently be expressed as $\Upsilon_N(x) = \frac{x_h}{x_l} \bmod N$.

Proposition 1. *Let* $x, y \in \mathbb{Z}$ *and co-prime to* N. *Then*

$$\Upsilon_N(x \cdot y) = \Upsilon_N(x) + \Upsilon_N(y) + \Upsilon_N(\underline{x} \cdot \underline{y}) \mod N$$

where $\underline{x} = x \bmod N$ *and* $\underline{y} = y \bmod N$.

Proof. Write $x \bmod N^2 = x_l + x_h N$ with $0 \leq x_l, x_h < N$ and $y \bmod N^2 = y_l + y_h N$ with $0 \leq y_l, y_h < N$. Note that $x_l = x \bmod N = \underline{x}$, $x_h = \lfloor (x \bmod N^2)/N \rfloor$, $y_l = y \bmod N = \underline{y}$ and $y_h = \lfloor (y \bmod N^2)/N \rfloor$. So, $x \cdot y \equiv x_l y_l + (x_l y_h + y_l x_h) N \equiv (x_l y_l \bmod N) + \big(x_l y_h + y_l x_h + \lfloor \frac{x_l y_l}{N} \rfloor \bmod N\big) N \pmod{N^2}$ and thus $\Upsilon_N(x \cdot y) \equiv \frac{x_l y_h + y_l x_h + \lfloor \frac{x_l y_l}{N} \rfloor}{x_l y_l} \equiv \frac{y_h}{y_l} + \frac{x_h}{x_l} + \frac{\lfloor \frac{x_l y_l}{N} \rfloor}{x_l y_l} \equiv \Upsilon_N(y) + \Upsilon_N(x) + \Upsilon_N(\underline{x} \cdot \underline{y}) \pmod{N}$, noting that $\underline{x} \cdot \underline{y} = x_l \cdot y_l = (x_l y_l \bmod N) + \lfloor \frac{x_l y_l}{N} \rfloor N$.

Corollary 1. *Let $x \in \mathbb{Z}$ and co-prime to N. Then $\Upsilon_N(x^{-1} \bmod N^2) = \Upsilon_N(2 - x \cdot x^\star)$ where $x^\star = x^{-1} \bmod N$.*

Proof. Through Hensel lifting, we have $x^{-1} \equiv x^\star(2 - x\, x^\star) \pmod{N^2}$; cf. [15, Lemma 3.1]. Hence, $\Upsilon_N(x^{-1} \bmod N^2) = \Upsilon_N(x^\star(2 - x\, x^\star) \equiv \Upsilon_N(x^\star) + \Upsilon_N(2 - x\, x^\star) + \Upsilon_N\big(x^\star \cdot (2 - x\, x^\star \bmod N)\big) \equiv 2\Upsilon_N(x^\star) + \Upsilon_N(2 - x\, x^\star) \equiv \Upsilon_N(2 - x\, x^\star) \pmod{N}$ since $x^\star < N$.

3 On-Line/Off-Line Encryption

3.1 Description

We present an efficient on-line/off-line encryption scheme using the Ups function. The on-line cost is only of one modular addition or, equivalently, one or two *integer* additions. The size of a ciphertext is of $2 \log_2 N$ bits.

An on-line/off-line encryption scheme

$\mathsf{KeyGen}(1^\kappa)$ Given a security parameter κ, the key generation algorithm generates two large primes p and q and forms the RSA-type modulus $N = pq$. The public key is $pk = N$ and the private key is $sk = (p, q)$. The message space is $\mathcal{M} = \mathbb{Z}/N\mathbb{Z}$.

$\mathsf{Enc}_{pk}(m)$ Let $m \in \mathcal{M}$ denote the message being encrypted under public key pk.

Off-line phase

- Pick uniformly at random an integer $r \overset{\$}{\leftarrow} [1, N)$ with $\gcd(r, N) = 1$ and compute $R = r^N \bmod N^2$;
- Form the coupon $(\mu, \nu) = \big(R \bmod N, \Upsilon_N(R)\big)$.

On-line phase

- Let $u = \mu$ and compute $v = (m + \nu) \bmod N$;
- Return the ciphertext $C = (u, v)$.

$\mathsf{Dec}_{sk}(C)$ Given a ciphertext $C = (u, v)$, the corresponding plaintext can be recovered using private key sk as

$$m = (v + \Upsilon_N(U)) \bmod N \qquad \text{with } U = u^{\lambda \cdot \lambda^\star} \bmod N^2$$

where $\lambda = \operatorname{lcm}(p - 1, q - 1)$ and $\lambda^\star = \lambda^{-1} \bmod N$.

It can be verified that decryption is correct. Indeed, if $C = (u, v)$ denotes the encryption of a message m—namely, $C = (u, v)$ where $u = R \bmod N$ and $v = m + \Upsilon_N(R) \bmod N$ with $R = r^N \bmod N^2$ for some integer $r \in [1, N)$ with $\gcd(r, N) = 1$, then $R^{\lambda \cdot \lambda^\star} \equiv (r^{\lambda^\star})^{N\lambda} \equiv 1 \pmod{N^2}$ by noting that $N\lambda$ is the exponent of the group $(\mathbb{Z}/N^2\mathbb{Z})^*$. Consequently, we get $\Upsilon_N(R^{\lambda\cdot\lambda^\star} \bmod N^2) = \Upsilon_N(1) = 0$. Further, from $R = (R \bmod N) + \lfloor R/N \rfloor N \equiv u(1 + \Upsilon_N(R)N) \pmod{N^2}$, we have

$$\begin{aligned}
0 &= \Upsilon_N(R^{\lambda\cdot\lambda^\star} \bmod N^2)\\
&= \Upsilon_N\big(u^{\lambda\cdot\lambda^\star}(1+\Upsilon_N(R)N)^{\lambda\cdot\lambda^\star} \bmod N^2\big)\\
&= \Upsilon_N\big(u^{\lambda\cdot\lambda^\star}(1+\Upsilon_N(R)N) \bmod N^2\big) && \text{since } \lambda\lambda^\star \equiv 1 \pmod{N}\\
&= \Upsilon_N(u^{\lambda\cdot\lambda^\star} \bmod N^2) + \underbrace{\Upsilon_N\big(1+\Upsilon_N(R)N\big)}_{=\frac{\Upsilon_N(R)}{1}} + \underbrace{\Upsilon_N(1\cdot 1)}_{=0} && \text{by Proposition 1}
\end{aligned}$$

modulo N. Hence, letting $U = u^{\lambda\cdot\lambda^\star} \bmod N^2$, we finally obtain $0 \equiv \Upsilon_N(U) + \Upsilon_N(R) \equiv \Upsilon_N(U) + (v - m) \pmod{N} \iff m = \Upsilon_N(U) + v \bmod N$.

Implementation notes Again, in practice, there is no need to check that $\gcd(r, N) = 1$. Note also that the evaluation of v does not really require a modular reduction since

$$(m + \nu) \bmod N = \begin{cases} m + \nu & \text{if } m + \nu < N\,,\\ m + \nu - N & \text{otherwise}\,. \end{cases}$$

When $x \equiv 1 \pmod{N}$, we have $\Upsilon_N(x^e) \equiv e \cdot \Upsilon_N(x) \pmod{N}$ for any exponent e. As a result, the evaluation of $\Upsilon_N(U)$ with $U = (u^\lambda)^{\lambda^\star} \bmod N^2$ can be carried out efficiently as $\Upsilon_N(U) = \lambda^\star \cdot \Upsilon_N(u^\lambda \bmod N^2) \bmod N$. Note also from the first property of the Ups function that $\Upsilon_N(x) = \frac{(x-1) \bmod N^2}{N}$ when $x \equiv 1 \pmod{N}$; hence, $\Upsilon_N(u^\lambda \bmod N^2) = \frac{(u^\lambda - 1) \bmod N^2}{N}$.

Further, decryption can be sped up through Chinese remaindering [16, § 14.5]: $m = \mathrm{CRT}(m_p, m_q)$ where $m_p = v + \Upsilon_p(U \bmod p^2) \bmod p$ and $m_q = v + \Upsilon_p(U \bmod q^2) \bmod q$.

Variants The above cryptosystem is subject to numerous variants. For example, one could define a ciphertext as a pair $(u^\star, v)$ with $u^\star = R^{-1} \bmod N$ and $v = (m + \Upsilon_N(R)) \bmod N$, where $R = r^N \bmod N^2$. Note that $\Upsilon_N(R) = \lfloor \frac{R \bmod N^2}{N} \rfloor u^\star \bmod N$.

3.2 Security Analysis

The security immediately follows from the security of Paillier cryptosystem. Indeed, a ciphertext (u, v) as per Section 3.1 can be converted into a regular Paillier ciphertext C as

$$C = u(1 + vN) \bmod N^2\ .$$

Conversely, a regular Paillier ciphertext C can be converted into an "on-line/off-line" ciphertext (u, v) where $u = C \bmod N$ and $v = L(C/u \bmod N^2) \bmod N$ with $L(x) = \frac{x-1}{N}$.

For completeness, security proofs are provided in Appendix B.

3.3 Homomorphic Operations

Addition The cryptosystem presented in Section 3.1 is additively homomorphic. That means that if C_1 and C_2 denote the respective encryptions of any two messages m_1 and m_2 in $\mathcal{M}$, there exists a publicly known operation, say $\boxplus$, such that the decryption algorithm returns message $m_1 + m_2$ (as an element of $\mathcal{M}$) on input ciphertext $C_1 \boxplus C_2$.

Specifically, the 'addition' of two ciphertexts, $C_1 = (u_1, v_1)$ and $C_2 = (u_2, v_2)$, is given by $C_3 \coloneqq C_1 \boxplus C_2 = (u_3, v_3)$ with

$$u_3 = u_1 u_2 \bmod N \quad \text{and} \quad v_3 = v_1 + v_2 + \varUpsilon_N(u_1 \cdot u_2) \pmod N \ . \tag{1}$$

This directly follows from Proposition 1. Consider two plaintexts $m_1, m_2 \in \mathcal{M}$. For $i \in \{1, 2\}$, write $R_i = {r_i}^N \bmod N^2$ with $r_i \xleftarrow{\$} [1, N)$, $u_i = R_i \bmod N$, $\nu_i = \varUpsilon_N(R_i)$, and $v_i = m_i + \nu_i \bmod N$. Then, defining $r_3 = r_1 r_2 \bmod N$ and $R_3 = {r_3}^N \bmod N^2$, we get $R_3 \equiv R_1 R_2 \equiv u_1 u_2 \pmod N$ and $\varUpsilon_N(R_3) \equiv \varUpsilon_N(R_1 R_2 \bmod N^2) \equiv \varUpsilon_N(R_1) + \varUpsilon_N(R_2) + \varUpsilon_N\big((R_1 \bmod N)(R_2 \bmod N)\big) \equiv \nu_1 + \nu_2 + \varUpsilon_N(u_1 u_2) \pmod N$. As a result, $u_3 \coloneqq R_3 \bmod N$ and $v_3 \coloneqq m_3 + \nu_3 \bmod N$ with $\nu_3 \coloneqq \varUpsilon_N(R_3)$ as per Equation (1) yield the encryption of message $m_3 \equiv v_3 - \nu_3 = (m_1 + \nu_1) + (m_2 + \nu_2) + \varUpsilon_N(u_1 u_2) - \varUpsilon_N(R_3) \equiv m_1 + m_2 \pmod N$.

Negation and subtraction In certain applications, when working over encrypted data, it is sometimes required to include negative numbers. When the message space $\mathcal{M}$ is (isomorphic to) the additive group $\mathbb{Z}/N\mathbb{Z}$, it is customary to view the elements of $\mathbb{Z}/N\mathbb{Z}$ as belonging to the set $\{-\lfloor N/2 \rfloor, \ldots, \lceil N/2 \rceil - 1\}$ in order to keep track of the sign. For the message space $\mathcal{M} = \{0, \ldots, N-1\}$, non-negative messages are represented by elements in $\{0, \ldots, \lceil N/2 \rceil - 1\}$ while negative messages by elements in $\{\lceil N/2 \rceil, \ldots, N-1\}$. So, the additive inverse of a message $m \in \mathcal{M}$ is given by $(-m \bmod N) = N - m$.

An application of the decryption algorithm to the ciphertext $(1, 0)$ produces plaintext $0 + \varUpsilon_N(1) = 0$. In other words, $(1, 0)$ corresponds to the encryption 0. Solving Equation (1) for (u_2, v_2) with $(u_3, v_3) = (1, 0)$ leads to $u_2 = {u_1}^{-1} \bmod N$ and $v_2 = -v_1 - \varUpsilon_N\big(u_1 \cdot ({u_1}^{-1} \bmod N)\big) \bmod N$. Therefore, the 'negation' of a ciphertext $C = (u, v)$, denoted by $\boxminus C = (u^\star, v^\star)$, can be obtained as

$$u^\star = u^{-1} \bmod N \quad \text{and} \quad v^\star = -v - \varUpsilon_N(u \cdot u^\star) \pmod N \ . \tag{2}$$

The negation operation gives rise to the 'subtraction' of ciphertexts. Given two ciphertexts $C_1 = (u_1, v_1)$ and $C_2 = (u_2, v_2)$, their subtraction is defined as

$C_4 \coloneqq C_1 \boxminus C_2 = C_1 \boxplus (\boxminus C_2) = (u_4, v_4)$ with

$$u_4 = u_1 {u_2}^\star \bmod N \quad \text{and} \quad v_4 = v_1 - v_2 - \varUpsilon_N(u_2 \cdot {u_2}^\star) + \varUpsilon_N(u_1 \cdot {u_2}^\star) \mod N \tag{3}$$

where ${u_2}^\star = {u_2}^{-1} \bmod N$.

Multiplication by a constant Yet another useful operation is the multiplication by a constant. Let $C = (u, v)$ be the encryption of a message $m \in \mathcal{M}$. Then, for a natural constant $k \in [0, N)$, the encryption of $m_k \coloneqq k \cdot m \pmod N \in \mathcal{M}$ is given by $C_k = (u_k, v_k) \coloneqq k \boxdot C = C \boxplus C \boxplus \cdots \boxplus C$ (k times) with

$$u_k = u^k \bmod N \quad \text{and} \quad v_k = kv + \varUpsilon_N(u^k \bmod N^2) \mod N \ . \tag{4}$$

This can be shown by induction. For $k = 0$, we have $m_0 = 0$ and Equation (4) yields $u_0 = 1$ and $v_0 = 0 \cdot v + \varUpsilon_N(1) \bmod N = 0$. Clearly, $(u_0, v_0) = (1, 0)$ is a valid encryption for message $m_0 = 0$. Now suppose that Equation (4) is valid for k; we have to prove that it remains valid for $k + 1$. Applying Equation (1) with $C_1 = (u_1, v_1)$ being the encryption of message m and $C_k = (u_k, v_k)$ that of message m_k, we get the encryption $C_{k+1} = (u_{k+1}, v_{k+1})$ of message m_{k+1} with $u_{k+1} \equiv u_1 u_k \equiv u u^k \equiv u^{k+1} \pmod N$ and $v_{k+1} \equiv v_1 + v_k + \varUpsilon_N(u_1 u_k) \equiv v + kv + \varUpsilon_N(u^k \bmod N^2) + \varUpsilon_N(u(u^k \bmod N)) \equiv (k + 1)v + \varUpsilon_N(u^{k+1} \bmod N^2) \pmod N$. The latter congruence follows from Proposition 1 by noting that $\varUpsilon_N(u^{k+1} \bmod N^2) \equiv \varUpsilon_N\big((u \bmod N^2)(u^k \bmod N^2)\big) \equiv \varUpsilon_N(u \bmod N^2) + \varUpsilon_N(u^k \bmod N^2) + \varUpsilon_N((u \bmod N)(u^k \bmod N)) \equiv 0 + \varUpsilon_N(u^k \bmod N^2) + \varUpsilon_N\big(u(u^k \bmod N)\big) \pmod N$.

Remark 2. Since $-1 \equiv N - 1 \pmod N$, Equation (4) yields an alternative way to get the negation of a ciphertext. If $C = (u, v)$ then $\big(u^{N-1} \bmod N, -v + \varUpsilon_N(u^{N-1} \bmod N^2) \bmod N\big)$ is also a valid expression for $\boxminus C$.

Re-randomization The additive homomorphism induced by $\boxplus$ enables the re-randomization of a ciphertext. This can be done by adding the encryption of 0 to a ciphertext. Specifically, if $C = (u, v)$ is the encryption of a message m, then $C^* = (u^*, v^*)$ with

$$u^* = u\varrho \bmod N \quad \text{and} \quad v^* = v + \varUpsilon_N(\varrho) + \varUpsilon_N\big(u \cdot (\varrho \bmod N)\big) \mod N$$

where $\varrho = \rho^N \bmod N^2$ for some $\rho \xleftarrow{\$} [1, N)$, is a randomized ciphertext which decrypts to the same message m.

This re-randomization step is important and must be applied to provide indistinguishability of encryptions.

4 Trapdoor Commitments

4.1 Generic Construction

Formally, a trapdoor commitment scheme consists of a tuple of three polynomial-time algorithms, $(\mathsf{KeyGen}, \mathsf{Com}, \mathsf{Open})$:

Key generation The key generation algorithm KeyGen is a probabilistic algorithm that takes on input a security parameter κ and outputs a pair of public and private key: $(pk, sk) \xleftarrow{\$} \mathsf{KeyGen}(1^\kappa)$.

Commitment Let $\mathcal{M}$ and $\mathcal{R}$ denote the "message" space and the randomness space, respectively. On input a value $m \in \mathcal{M}$, the commitment function Com draws at random $\rho \xleftarrow{\$} \mathcal{R}$, computes commitment C using public key pk, and returns C. We write $C \leftarrow \mathsf{Com}_{pk}(m; \rho)$.

Opening The opening function Open takes on input a commitment C and a value $m \in \mathcal{M}$. It returns a value $\rho' \in \mathcal{R}$ using private key sk (matching pk). We write $\rho' \leftarrow \mathsf{Open}_{sk}(C, m)$.

Correctness requires that for all $(pk, sk) \xleftarrow{\$} \mathsf{KeyGen}(1^\kappa)$,

$$\mathsf{Com}_{pk}(m; \rho') = C$$

for any value $m \in \mathcal{M}$, any commitment thereto $C \leftarrow \mathsf{Com}_{pk}(m; \rho)$ with $\rho \xleftarrow{\$} \mathcal{R}$, and $\rho' \leftarrow \mathsf{Open}_{sk}(C, m)$. For security, we need the following properties:

1. Hiding property: For all probabilistic polynomial adversaries $\mathcal{A}$,

$$\left| \Pr\left[b' = b \;\middle|\; \begin{array}{l} (pk, sk) \xleftarrow{\$} \mathsf{KeyGen}(1^\kappa); (m_0, m_1) \in \mathcal{M}^2 \leftarrow \mathcal{A}(pk); \\ b \xleftarrow{\$} \{0, 1\}; \rho \xleftarrow{\$} \mathcal{R}; C^* \leftarrow \mathsf{Com}_{pk}(m_b; \rho); b' \leftarrow \mathcal{A}(pk, C^*) \end{array} \right] - \frac{1}{2} \right|$$

is negligible in κ;

2. Binding property: For all probabilistic polynomial adversaries $\mathcal{A}$,

$$\Pr\Bigg[\mathsf{Com}_{pk}(m_0; \rho_0) = \mathsf{Com}_{pk}(m_1; \rho_1) \wedge m_0 \neq m_1 \;\Bigg|\; \begin{array}{l} (pk, sk) \xleftarrow{\$} \mathsf{KeyGen}(1^\kappa); \\ (m_0, \rho_0), (m_1, \rho_1) \in \mathcal{M} \times \mathcal{R} \leftarrow \mathcal{A}(pk) \end{array} \Bigg]$$

is negligible in κ.

An abstract scheme Let $(\overline{\mathsf{KeyGen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}})$ be an homomorphic encryption scheme with recoverable randomness.[2] We assume that the message space is an additive group $\mathcal{M}_o \cong \mathbb{Z}/N\mathbb{Z}$ and let ${\mathcal{M}_o}^*$ denote the set of invertible elements; the randomness space is denoted by $\mathcal{R}_o$. Let $(pk_o, sk_o) \xleftarrow{\$} \overline{\mathsf{KeyGen}}(1^\kappa)$. In order

[2] That is, where the randomness used during encryption can be recovered together with the message by the decryption algorithm.

to capture the probabilistic nature of the encryption, we explicitly include the randomness in the encryption algorithm and write $C \leftarrow \overline{\mathsf{Enc}}_{pk_o}(m, r)$ for the encryption of $m \in \mathcal{M}_o$ with randomness $r \in \mathcal{R}_o$. Also, we suppose that the decryption algorithm returns both the plaintext and the used randomness; we write $(m, r) \leftarrow \overline{\mathsf{Dec}}_{sk_o}(C)$. We use $\boxplus$, $\boxminus$ and $\boxdot$ for operations on ciphertexts; see Section 3.3.

A trapdoor commitment scheme $(\mathsf{KeyGen}, \mathsf{Com}, \mathsf{Open})$ can be obtained as follows.

$\mathsf{KeyGen}(1^\kappa)$ 1. Run $\overline{\mathsf{KeyGen}}(1^\kappa)$ and obtain $(pk_o, sk_o) \overset{\$}{\leftarrow} \overline{\mathsf{KeyGen}}(1^\kappa)$;
2. Draw $\mu_o \overset{\$}{\leftarrow} {\mathcal{M}_o}^*$ and $r_o \overset{\$}{\leftarrow} \mathcal{R}_o$, and compute $C_o \leftarrow \overline{\mathsf{Enc}}_{pk_o}(\mu_o, r_o)$;
3. Output $pk = (pk_o, C_o)$ and $sk = (sk_o, \mu_o)$.

The message space is $\mathcal{M} := \mathcal{M}_o$ and the randomness space is $\mathcal{R} := \mathcal{R}_o \times \mathcal{M}_o$.

$\mathsf{Com}_{pk}(m; (r, s))$ Given message $m \in \mathcal{M}$ and randomness $\rho := (r, s) \overset{\$}{\leftarrow} \mathcal{R}$, return

$$C \leftarrow \overline{\mathsf{Enc}}_{pk_o}(m, r) \boxplus (s \boxdot C_o) \enspace .$$

$\mathsf{Open}(C, m)$ 1. Compute $(m', \cdot) \leftarrow \overline{\mathsf{Dec}}_{sk_o}(C)$ and $s' \leftarrow (m' - m){\mu_o}^{-1}$ $(\in \mathcal{M})$;
2. Compute $C' \leftarrow C \boxminus (s' \boxdot C_o)$ and $(\cdot, r') \leftarrow \overline{\mathsf{Dec}}_{sk_o}(C')$;
3. Return $\rho' = (r', s')$.

It can be verified that the scheme is correct, namely that the value $\rho' = (r', s') \leftarrow \mathsf{Open}(C, m)$ is accepting w.r.t. commitment C and message $m \in \mathcal{M}$. We need to show that if $C \leftarrow \mathsf{Com}_{pk}(m; (r, s)) = \overline{\mathsf{Enc}}_{pk_o}(m, r) \boxplus (s \boxdot C_o)$ then $(r', s') = (r, s)$. This follows from the fact that the encryption function $\overline{\mathsf{Enc}}_{pk_o} \colon \mathcal{R} \to \mathcal{R}$ (with $\mathcal{R} = \mathcal{R}_o \times \mathcal{M}_o$) is one-to-one. Indeed, we have $C = \overline{\mathsf{Enc}}_{pk_o}(m, r) \boxplus (s \boxdot C_o) = \overline{\mathsf{Enc}}_{pk_o}(m + s \cdot \mu_o, r'')$ for some $r'' \in \mathcal{R}_o$. Hence, letting $m' := m + s \cdot \mu_o$, we get $s' \leftarrow (m' - m){\mu_o}^{-1} = s$. In turn, letting $C' := C \boxminus (s' \boxdot C_o) = \overline{\mathsf{Enc}}_{pk_o}(m'', r')$ for some $m'' \in \mathcal{M}_o$, we get $C' = C \boxminus (s' \boxdot C_o) = C \boxminus (s \boxdot C_o) = \overline{\mathsf{Enc}}_{pk_o}(m, r)$ and thus $r' \leftarrow \overline{\mathsf{Dec}}_{sk_o}(C')[2] = r$.

Regarding the security, the scheme is perfectly hiding. Indeed, the sole information an adversary $\mathcal{A}$ can get on random bit b in the security game (cf. Appendix B) is from $C^* \leftarrow \overline{\mathsf{Enc}}_{pk_o}(m_b, r) \boxplus (s \boxdot C_o)$ where $(r, s) \overset{\$}{\leftarrow} \mathcal{R}$. But C^* is an encryption of $m^* := m_b + s \cdot \mu_o$ and m^* is uniformly distributed over $\mathcal{M}$ since $s \overset{\$}{\leftarrow} \mathcal{M}$. So the best $\mathcal{A}$ can do is to return at random $b' \in \{0, 1\}$ as its guess for the value of b.

The scheme is also binding under the assumption that the encryption scheme $\overline{\mathsf{Enc}}$ is one-way. By contradiction, suppose that there exists an efficient algorithm $\mathcal{A}$ that, on input $pk = (pk_o, C_o)$ where $C_o \leftarrow \overline{\mathsf{Enc}}_{pk_o}(\mu_o, r_o)$ with $r_o \overset{\$}{\leftarrow} \mathcal{R}_o$, can find two colluding pairs $(m_0, \rho_0), (m_1, \rho_1) \in \mathcal{M} \times \mathcal{R}$ with $m_0 \neq m_1$, where $\rho_0 = (r_0, s_0)$ and $\rho_1 = (r_1, s_1)$. This means that $\mathsf{Com}_{pk}(m_0; (r_0, s_0)) = \mathsf{Com}_{pk}(m_1; (r_1, s_1)) \iff \overline{\mathsf{Enc}}_{pk_o}(m_0, r_0) \boxplus (s_0 \boxdot C_o) = \overline{\mathsf{Enc}}_{pk_o}(m_1, r_1) \boxplus (s_1 \boxdot C_o)$ with $C_o = \overline{\mathsf{Enc}}_{pk_o}(\mu_o, r_o)$. As a consequence, since $\overline{\mathsf{Enc}}$ is one-to-one, we must have $m_0 + s\mu_o = m_1 + s_1\mu_o \iff (s_0 - s_1)\mu_o = m_1 - m_0$ (as elements in $\mathcal{M}_o$) and thus $\mathcal{A}$ can recover μ_o—remember that $m_0 \neq m_1$ and so $s_0 \neq s_1$ since $\mu_o \in {\mathcal{M}_o}^*$.

4.2 On-line/Off-line Trapdoor Commitments

Specializing the previous abstract scheme to the encryption of Section 3.1 yields a trapdoor commitment scheme that requires only one *modular addition* (or, equivalently, one or two integer additions) in the on-line-phase. This has to be compared with state-of-the-art on-line/off-line trapdoor commitment schemes of [7] and [6] that involve modular multiplications.

A trapdoor commitment scheme

$\mathsf{KeyGen}(1^\kappa)$ Given a security parameter κ, the key generation algorithm generates two large primes p and q and forms the RSA-type modulus $N = pq$. The message space is $\mathcal{M} = \{0, 1, 2, \ldots, N-1\}$ and the randomness space is $\mathcal{R} = \mathcal{M}^* \times \mathcal{M}$. The algorithm also computes $R_o = {r_o}^N \bmod N^2$ for some $r_o \xleftarrow{\$} \mathcal{M}^*$ and sets $u_o = R_o \bmod N$ and $v_o = (\mu_o + \Upsilon_N(R_o)) \bmod N$ with $\mu_o \xleftarrow{\$} \mathcal{M}^*$. The public key is $pk = (N, u_o, v_o)$ and the private key is $sk = (p, q, \mu_o)$.

$\mathsf{Com}_{pk}(m; (r, s))$ Let $m \in \mathcal{M}$ denote the message being committed to under public key pk.

Off-line phase

- Pick uniformly at random $(r, s) \xleftarrow{\$} \mathcal{R}$ and compute $W = {u_o}^s\, r^N \bmod N^2$;
- Form the coupon (μ, ν) as $(\mu, \nu) = \big(W \bmod N, (\Upsilon_N(W) + s\, v_o) \bmod N\big)$.

On-line phase

- Let $u = \mu$ and compute $v = (m + \nu) \bmod N$;
- Return the commitment $C = (u, v)$.

$\mathsf{Open}_{sk}(C, m)$ A commitment $C = (u, v)$ to a message $m \in \mathcal{M}$ can be open using private key sk by letting $U = u^{\lambda \cdot \lambda^\star} \bmod N^2$ and returning the pair (r', s') satisfying

$$s' = \frac{v + \Upsilon_N(U) - m}{\mu_o} \bmod N \quad \text{and} \quad r' = \big(u\, {u_o}^{-s'}\big)^{N^\star} \bmod N$$

where $\lambda \coloneqq \lambda(N) = \operatorname{lcm}(p-1, q-1)$, $\lambda^\star = \lambda^{-1} \bmod N$, and $N^\star = N^{-1} \bmod \lambda$.

Variants Again, many variants are possible. For example, the private key could include ${\mu_o}^{-1} \bmod N$ (instead of μ_o) to avoid dividing by μ_o.

4.3 Chameleon Signatures

Regular digital signatures offer *non-repudiation* in addition to authenticity. This additional property is sometimes undesired. Chameleon signatures are recipientspecific: the signature's recipient can authenticate a signed message but has no way to convince a third party that the message originated from the signer.

The construction is fairly simple. If (pk_R, sk_R) denote the recipient's key pair for a non-interactive trapdoor commitment scheme $(\mathsf{KeyGen}, \mathsf{Com}, \mathsf{Open})$, then to chameleon-sign a message $m \in \mathcal{M}$, the signer

- chooses $\rho \xleftarrow{\$} \mathcal{R}$;
- forms the "augmented message" $\hat{m} = \mathcal{G}\big(\mathsf{Com}_{pk_R}(m;\rho), pk_R\big)$ where $\mathcal{G}$ is a collision-resistant hash function;[3] and
- computes the signature on $\hat{m}$.

Clearly, the so-obtained signature is not transferable to a third party since the recipient is able with private key sk_R to find randomness $\rho' \in \mathcal{R}$ for any *chosen* message $m' \in \mathcal{M}$ such that $\hat{m} = \big(\mathsf{Com}_{pk_R}(m';\rho'), pk_R\big)$. In other words, for everyone but the recipient, the signature could be the signature on any message m'.

Key-exposure freeness There is a subtle issue with chameleon signatures: key exposure. As shown in the proof of the binding property (cf. Section 4.1), a collision forgery results in the signer recovering the value of μ_o from two colliding pairs (r_0, s_0) and (r_1, s_1), respectively committing to two distinct messages m_0 and m_1, as $\mu_o = (m_1 - m_0)/(s_0 - s_1)$.

Remark 3. With the scheme of Section 4.2, the signer is even able to recover the randomness that was used to encrypt μ_o. Since (r_0, s_0) and (r_1, s_1) are colliding, we have ${u_o}^{s_0}{r_0}^N \equiv {u_o}^{s_1}{r_1}^N \pmod{N} \iff {u_o}^{s_0-s_1} \equiv (r_1/r_0)^N \pmod{N}$. An application of the extended Euclidean algorithm to $(s_0 - s_1, N)$ gives two integers α and β such that $\alpha(s_0 - s_1) + \beta N = \gcd(s_0 - s_1, N) = 1$. As a consequence, we get $u_o \equiv {u_o}^{\alpha(s_0-s_1)+\beta N} \equiv \big((r_1/r_0)^\alpha {u_o}^\beta\big)^N \pmod{N}$ and thus $r_o = (r_1/r_0)^\alpha {u_o}^\beta \bmod N$. Now, using r_o, the signer is able to compute chosen collisions and can therefore deny *other* signatures given to the recipient. Indeed, given (m, r, s), if $\mathsf{Com}_{pk_R}(m;(r,s)) = C$, then for any chosen message m', $\mathsf{Com}_{pk_R}(m';(r',s') = C$ by letting $s' := s + {\mu_o}^{-1}(m - m')$ and $r' := {r_o}^{s-s'} r \bmod N$.

In order to address this limitation, we make μ_o dependent on the transaction, say τ, in chameleon signatures by

1. appending a "label" $\ell := \ell(\tau)$ in the augmented message; i.e.,
$$\hat{m} = \mathcal{G}\big(\mathsf{Com}_{pk_R}(m;(r,s)), pk_R, \ell(\tau)\big)\;;$$
2. defining (u_o, v_o) as $(u_o, v_o) := \big(u_o(\tau), v_o(\tau)\big) = \mathcal{H}\big(\ell(\tau)\big)$ where $\mathcal{H}$ is a cryptographic hash function mapping to $(\mathbb{Z}/N\mathbb{Z})^* \times \mathbb{Z}/N\mathbb{Z}$, viewed as a random oracle [4].

[3] As noted in [14, § 4.2], it is important to append pk_R (along with a description of the chameleon hash function Com) in the evaluation of augmented message $\hat{m}$. Otherwise, the signer or the recipient could claim that the chameleon hash was generated under a different hash function.

The corresponding value for μ_o is therefore implicitly defined as

$$\mu_o \coloneqq \mu_o(\tau) = v_o(\tau) - \Upsilon_N\bigl(R_o(\tau)\bigr)$$

with $R_o(\tau) = r_o(\tau)^N \bmod N^2$ where $r_o(\tau) = u_o(\tau)^{1/N} \bmod N$. The label can be seen as a unique transaction identifier.

The property of key-exposure freeness is easily verified. If the signer were able to find a collision for the target transaction τ^* with label $\ell(\tau^*)$ then, similarly to Remark 3, she could recover $r_o(\tau^*) = u_o(\tau^*)^{1/N} \bmod N$; that is, an N^{th} root modulo N. This means inverting the RSA function with exponent N. Note also the public-key components $\bigl\{(u_o(\tau), v_o(\tau))\bigr\}_\tau$ are uniformly distributed.

Finally, we observe that using an on-line/off-line scheme (e.g., [13]) for the signature step leads to an on-line/off-line chameleon signature scheme.

5 Conclusion

In this paper, we have proposed an efficient on-line/off-line DCR-based homomorphic encryption scheme and companion trapdoor commitment scheme. Both schemes just require *one or two integer additions* in their on-line phase. The on-line efficiency makes the proposals particularly well suited to time-constrained applications or to low-end devices that do not have much computational resources.

A Public-Key Encryption

A *public-key encryption scheme* (see e.g. [16, Chapter 8]) is a tuple of three polynomial-time algorithms, $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$:

Key generation The key generation algorithm KeyGen is a probabilistic algorithm that takes on input a security parameter κ and outputs a pair of public and private key: $(pk, sk) \xleftarrow{\$} \mathsf{KeyGen}(1^\kappa)$.

Encryption Let $\mathcal{M}$ denote the message space. The encryption algorithm Enc is a randomized algorithm that takes on input a public key pk and a plaintext $m \in \mathcal{M}$, and returns a ciphertext C. We write $c \leftarrow \mathsf{Enc}_{pk}(m)$.

Decryption The decryption algorithm Dec takes on input secret key sk (matching pk) and ciphertext C. It returns the corresponding plaintext m or a special symbol $\perp$ indicating that the ciphertext is invalid. We write $m \leftarrow \mathsf{Dec}_{sk}(C)$ if C is a valid ciphertext and $\perp \leftarrow \mathsf{Dec}_{sk}(C)$ if it is not.

It is required that for all $(pk, sk) \xleftarrow{\$} \mathsf{KeyGen}(1^\kappa)$, $\mathsf{Dec}_{sk}\bigl(\mathsf{Enc}_{pk}(m)\bigr) = m$ for any message $m \in \mathcal{M}$.

B Security Proofs

B.1 One-Wayness

One-wayness is the minimal security requirement an encryption scheme must meet: An adversary should not be able to recover the plaintext given its encryption.

The cryptosystem of Section 3.1 fulfills this requirement under the Hensel Lifting assumption [8].

Assumption 1 (Hensel Lifting). *Let κ be a security parameter. Let also* $\mathsf{RSAgen}(1^\kappa)$ *be a probabilistic polynomial-time algorithm that generates two equal-size primes p and q. The* Composite Residuosity assumption *conjectures that for all probabilistic polynomial-time algorithms $\mathcal{B}$,*

$$\Pr\bigl[\mathcal{B}(N,y) = x^N \bmod N^2 \mid (p,q) \xleftarrow{\$} \mathsf{RSAgen}(1^\kappa); N \leftarrow pq; \\ x \xleftarrow{\$} (\mathbb{Z}/N\mathbb{Z})^*; y \leftarrow x^N \bmod N\bigr]$$

is negligible in κ.

The proof is by reduction. We assume that there exists an adversary $\mathcal{A}$ against the one-wayness property of the scheme. We will use this adversary to break the Hensel Lifting assumption. Consider the following algorithm $\mathcal{B}$ receiving as an input a challenge $(\hat{N}, \hat{y})$ where $\hat{N} \xleftarrow{\$} \mathsf{RSAgen}(1^\kappa)$ and $\hat{y} = \hat{x}^N \bmod N$ with $\hat{x} \xleftarrow{\$} (\mathbb{Z}/N\mathbb{Z})^*$:

1. $\mathcal{B}$ sets $N = \hat{N}$ and defines $pk = N$. It also sets $u = \hat{y}$, draws $v \xleftarrow{\$} \{0,1,\ldots,N-1\}$, and lets $C = (u,v)$. It gives public key pk and challenge ciphertext C to $\mathcal{A}$.
2. $\mathcal{A}$ returns a plaintext m—remark here that all ciphertexts are valid.
3. From the received m, $\mathcal{B}$ outputs $Y \coloneqq u + Nu(v-m) \pmod{N^2}$.

Observe that $u = \hat{x}^N \bmod N$ and, if $m = \mathsf{Dec}_{sk}(C)$, that $v - m \equiv \varUpsilon_N(\hat{x}^N \bmod N^2) \pmod N$. As a result, we have $Y \equiv (\hat{x}^N \bmod N) + N\left\lfloor \frac{\hat{x}^N \bmod N^2}{N} \right\rfloor \equiv \hat{x}^N \pmod{N^2}$.

In turn, as shown in [8, Theorem 2], we get that the one-wayness of the cryptosystem holds under the Computational Composite Residuosity (CCR) assumption.

Assumption 2 (Computational Composite Residuosity [17]). *Let κ be a security parameter and let* $\mathsf{RSAgen}(1^\kappa)$ *be a probabilistic polynomial-time algorithm that generates two equal-size primes p and q. The* CCR assumption *conjectures that for all probabilistic polynomial-time algorithms $\mathcal{B}$,*

$$\Pr\left[\mathcal{B}(N,y,g) = c \;\middle|\; \begin{array}{l} (p,q) \xleftarrow{\$} \mathsf{RSAgen}(1^\kappa); N \leftarrow pq; \\ g \xleftarrow{\$} (\mathbb{Z}/N^2\mathbb{Z})^* \; s.t. \; \mathrm{ord}(g) \propto N; c \xleftarrow{\$} \{0,1,\ldots,N-1\}; \\ x \xleftarrow{\$} (\mathbb{Z}/N^2\mathbb{Z})^*; y \leftarrow g^c x^N \bmod N^2 \end{array}\right]$$

is negligible in κ.

B.2 Semantic Security

We now show that the cryptosystem of Section 3.1 is semantically secure [12] under the Decisional Composite Residuosity (DCR) assumption.

Assumption 3 (Decisional Composite Residuosity [17]). *Let κ be a security parameter and let $\mathsf{RSAgen}(1^\kappa)$ be a probabilistic polynomial-time algorithm that generates two equal-size primes p and q. Consider the distributions $\mathrm{dist}_0(\kappa)$ and $\mathrm{dist}_1(\kappa)$ given by*

$$\mathrm{dist}_0(\kappa) = \bigl\{(N,R) \mid N \leftarrow pq \textit{ with } (p,q) \xleftarrow{\$} \mathsf{RSAgen}(1^\kappa) \wedge R \xleftarrow{\$} (\mathbb{Z}/N^2\mathbb{Z})^*\bigr\}$$

and

$$\mathrm{dist}_1(\kappa) = \bigl\{(N,R) \mid N \leftarrow pq \textit{ with } (p,q) \xleftarrow{\$} \mathsf{RSAgen}(1^\kappa) \wedge R \leftarrow r^N \bmod N^2 \textit{ with } r \xleftarrow{\$} (\mathbb{Z}/N^2\mathbb{Z})^*\bigr\} \enspace .$$

The DCR assumption *conjectures that for all probabilistic polynomial-time algorithms $\mathcal{B}$, the function*

$$\Bigl|\Pr\bigl[\mathcal{B}(N,R)=1 \mid (N,R) \xleftarrow{\$} \mathrm{dist}_0(\kappa)\bigr] - \Pr\bigl[\mathcal{B}(N,R)=1 \mid (N,R) \xleftarrow{\$} \mathrm{dist}_1(\kappa)\bigr]\Bigr|$$

is negligible in κ.

The semantic security game between a challenger $\mathcal{B}$ and an adversary $\mathcal{A}$ proceeds as follows. The challenger is given a DCR challenge $(N,R) \xleftarrow{\$} \mathrm{dist}_\beta(\kappa)$ with $\beta \xleftarrow{\$} \{0,1\}$. Its goal is to tell if $\beta = 0$ or $\beta = 1$. For this purpose, $\mathcal{B}$ has access to adversary $\mathcal{A}$. The advantage of $\mathcal{A}$ in breaking the semantic security of the cryptosystem (i.e., to correctly recover b) is denoted by $\mathrm{adv}_{\mathcal{A}}^{\text{IND-CPA}}(\kappa)$. We need to show that this advantage is negligible.

Suppose that $\mathcal{B}$ runs as follows:

1. $\mathcal{B}$ sets the public key $pk = N$ and gives it to $\mathcal{A}$.
2. Let $\mathcal{M} = \{0,\dots,N-1\}$. $\mathcal{A}$ selects a pair of equal-length messages $m_0, m_1 \in \mathcal{M}$, $m_0 \neq m_1$.
3. $\mathcal{B}$ chooses at random $b \xleftarrow{\$} \{0,1\}$ and returns to $\mathcal{A}$ the challenge ciphertext $C^* := \bigl(R \bmod N, (m_b + \varUpsilon_N(R)) \bmod N\bigr)$ as the encryption of m_b.
4. $\mathcal{A}$ returns its guess $b' \in \{0,1\}$ that C^* is the encryption of $m_{b'}$.
5. $\mathcal{B}$ outputs 1 if $b' = b$, and 0 otherwise.

There are two cases to consider:

Case I: $(N,R) \in \mathrm{dist}_0(\kappa)$. In this case, R is uniform over $(\mathbb{Z}/N^2\mathbb{Z})^*$. As a consequence, $u^* := R \bmod N$ is a uniformly random value in $(\mathbb{Z}/N\mathbb{Z})^*$ and $v^* := (m_b + \varUpsilon_N(R)) \bmod N$ is a uniformly random value in $\mathbb{Z}/N\mathbb{Z}$ since $\varUpsilon_N(R)$ is uniform over $\mathbb{Z}/N\mathbb{Z}$. Message m_b is therefore completely hidden from the view of $\mathcal{A}$. Hence, we get $\Pr[\mathcal{B}(N,R)=1] = \frac{1}{2}$.

Case II: $(N, R) \in \mathrm{dist}_1(\kappa)$**.** In this case, $\mathcal{B}$ perfectly emulates the semantic security game. Indeed, we have $R = r^N \bmod N^2$ with $r \leftarrow (\mathbb{Z}/N^2\mathbb{Z})^*$, which is equivalent to $R = \underline{r}^N \bmod N^2$ where $\underline{r} \coloneqq r \bmod N$ satisfies $\underline{r} \in [1, N)$ and $\gcd(\underline{r}, N) = 1$. We so get

$$\left|\Pr[\mathcal{B}(N,R)=1]-\tfrac{1}{2}\right| = \left|\Pr[b'=b]-\tfrac{1}{2}\right| = \mathrm{adv}_{\mathcal{A}}^{\text{IND-CPA}}(\kappa) \ .$$

Under the DCR assumption, we know that $\mathcal{B}$ cannot distinguish $\mathrm{dist}_0(\kappa)$ from $\mathrm{dist}_1(\kappa)$—with non-negligible probability. Combining the above two cases, we so deduce that

$$\begin{aligned}
\mathrm{adv}_{\mathcal{A}}^{\text{IND-CPA}}(\kappa) &= \left|\Pr\left[\mathcal{B}(N,R)=1 \mid (N,R) \overset{\$}{\leftarrow} \mathrm{dist}_1(\kappa)\right] - \tfrac{1}{2}\right| \\
&= \Bigg|\left(\Pr\left[\mathcal{B}(N,R)=1 \mid (N,R) \overset{\$}{\leftarrow} \mathrm{dist}_1(\kappa)\right] - \tfrac{1}{2}\right) - \\
&\qquad\qquad \left(\overbrace{\Pr\left[\mathcal{B}(N,R)=1 \mid (N,R) \overset{\$}{\leftarrow} \mathrm{dist}_0(\kappa)\right] - \tfrac{1}{2}}^{=0 \ \ \text{(Case I)}}\right)\Bigg| \\
&= \Big|\Pr\left[\mathcal{B}(N,R)=1 \mid (N,R) \overset{\$}{\leftarrow} \mathrm{dist}_0(\kappa)\right] - \\
&\qquad\qquad \Pr\left[\mathcal{B}(N,R)=1 \mid (N,R) \overset{\$}{\leftarrow} \mathrm{dist}_1(\kappa)\right]\Big| \\
&= \mathrm{negl}(\kappa) \ .
\end{aligned}$$

References

1. Ateniese, G., de Medeiros, B.: Identity-based chameleon hash and applications. In: Juels, A. (ed.) Financial Cryptography (FC 2004). Lecture Notes in Computer Science, vol. 3110, pp. 164–180. Springer (2004). https://doi.org/10.1007/978-3-540-27809-2_19
2. Ateniese, G., de Medeiros, B.: On the key exposure problem in chameleon hashes. In: Blundo, C., Cimato, S. (eds.) Security in Communication Networks (SCN 2004). Lecture Notes in Computer Science, vol. 3352, pp. 165–179. Springer (2004). https://doi.org/10.1007/978-3-540-30598-9_12
3. Bellare, M., Ristov, T.: A characterization of chameleon hash functions and new, efficient designs. Journal of Cryptology **27**(4), 799–823 (2014). https://doi.org/10.1007/s00145-013-9155-8
4. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Denning, D.E., et al. (eds.) 1st Conference on Computer and Communications Security (ACM CCS 1993). pp. 62–73. ACM Press (1993). https://doi.org/10.1145/168588.168596
5. Brassard, G., Chaum, D., Crépeau, C.: Minimum disclosure proofs of knowledge. Journal of Computer and System Sciences **37**(2), 156–189 (1988). https://doi.org/10.1016/0022-0000(88)90005-0
6. Bresson, E., Catalano, D., Pointcheval, D.: A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications. In: Laih, C.S. (ed.) Advances in Cryptology – ASIACRYPT 2003. Lecture Notes in Computer Science, vol. 2894, pp. 37–54. Springer (2003). https://doi.org/10.1007/978-3-540-40061-5_3

7. Catalano, D., Gennaro, R., Howgrave-Graham, N., Nguyen, P.Q.: Paillier's cryptosystem revisited. In: Reiter, M.K., Samarati, P. (eds.) 8th Conference on Computer and Communications Security (ACM CCS 2001). pp. 206–214. ACM Press (2001). https://doi.org/10.1145/501983.502012
8. Catalano, D., Nguyen, P.Q., Stern, J.: The hardness of Hensel lifting: The case of RSA and discrete logarithm. In: Zheng, Y. (ed.) Advances in Cryptology – ASIACRYPT 2002. Lecture Notes in Computer Science, vol. 2501, pp. 299–310. Springer (2002). https://doi.org/10.1007/3-540-36178-2_19
9. Chen, X., Zhang, F., Kim, K.: Chameleon hashing without key exposure. In: Zhang, K., Zheng, Y. (eds.) Information Security (ISC 2004). Lecture Notes in Computer Science, vol. 3225, pp. 87–98. Springer (2004). https://doi.org/10.1007/978-3-540-30144-8_8
10. Even, S., Goldreich, O., Micali, S.: On-line/off-line digital signatures. Journal of Cryptology **9**(1), 35–67 (1996). https://doi.org/10.1007/BF02254791
11. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. Journal of the ACM **38**(3), 690–728 (1991). https://doi.org/10.1145/116825.116852
12. Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences **28**(2), 270–299 (1984). https://doi.org/10.1016/0022-0000(84)90070-9
13. Joye, M.: An efficient on-line/off-line signature scheme without random oracles. In: Franklin, M.K., Hui, L.C.K., Wong, D.S. (eds.) Cryptology and Network Security (CANS 2008). Lecture Notes in Computer Science, vol. 5339, pp. 98–107. Springer (2008). https://doi.org/10.1007/978-3-540-89641-8_7
14. Krawczyk, H., Rabin, T.: Chameleon signatures. In: Network and Distributed System Security Symposium (NDSS 2000). The Internet Society (2000), https://www.ndss-symposium.org/ndss2000/chameleon-signatures/
15. Kurosawa, K., Takagi, T.: One-wayness equivalent to general factoring. IEEE Transactions on Information Theory **55**(9), 4249–4262 (2009). https://doi.org/10.1109/TIT.2009.2025532
16. Menezes, A.J., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press (1997). https://doi.org/10.1201/9780429466335
17. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) Advances in Cryptology – EUROCRYPT '99. Lecture Notes in Computer Science, vol. 1592, pp. 223–238. Springer (1999). https://doi.org/10.1007/3-540-48910-X_16