Reducing the Ciphertext Size of Dolev-Dwork-Naor like Public Key Cryptosystems

Rafael Dowsley[1], Goichiro Hanaoka[2] Hideki Imai[2], Anderson C. A. Nascimento[1]

[1] Department of Electrical Engineering, University of Brasilia
Campus Darcy Ribeiro, 70910-900, Brasilia, DF, Brazil
E-mail: rafaeldowsley@redes.unb.br, andclay@ene.unb.br
[2] National Institute of Advanced Industrial Science and Technology (AIST)
1-18-13, Sotokanda, Chyioda-ku, 101-0021, Tokyo, Japan
E-mail: {hanaoka-goichiro, h-imai}@aist.go.jp

Abstract. We show a method for compressing the ciphertext and reducing the computational cost of the Dolev-Dwork-Naor cryptosystem and related schemes without changing their other parameters nor reducing the original security levels.

1 Introduction

Indistinguishability of messages under adaptive chosen ciphertext attacks [11] is the strongest known notion of security for public key encryption schemes (PKE). In this notion of security, the adversary is given access to a decryption oracle. Many computational assumptions have been used in the literature for obtaining cryptosystems meeting such a strong security requirements and the study of generic constructions of public key encryption schemes meeting this security notion based on weak primitives is an interesting topic. Given enhanced one-way trapdoor permutations, we know how to obtain CCA2 security from any semantically secure public key cryptosystem [8, 1, 13, 7]. It is also possible to obtain CCA2 secure PKE based on correlated products [12] (which can be constructed based on certain lossy trapdoor functions [10]).

In this short note, we show a method for compressing a ciphertext of the Dolev-Dwork-Naor (DDN) [1] cryptosystem to half of its size. Our scheme also reduces the computational cost of the cryptosystem.

1.1 Extensions

Even tough we present proofs of security just for the original DDN scheme, our ideas can be straightforwardly applied to other CCA2 secure constructions which use many pairs of public/private keys. For instance, we can also short en the ciphertext and computational costs of the following schemes: the Rosen-Segev construction [12]; the Pass-Shelat-Vaikuntanathan construction [9]; the Dowsley-Müller-Quade-Nascimento construction [2] and the Hanaoka-Imai-Ogawa-Watanabe construction [4].

1.2 Post-Quantum Schemes

Reducing the ciphertext size of the original DDN construction is a rather theoretical contribution, as practical schemes are usually not based on this general reduction. However, in the case of post-quantum schemes, all the known protocols achieving CCA2 security in the standard model are based on general reduction techniques [12, 9, 2]. Thus, reducing their communication and computational complexities is indeed an important issue towards obtaining close to practical post-quantum public key schemes with CCA2 security in the standard model.

2 Preliminaries

2.1 Notation

If x is a string, then $|x|$ denotes its length, while $|S|$ represents the cardinality of a set S. If $k \in \mathbb{N}$, then 1^k denotes the string of k ones. $s \leftarrow S$ denotes the operation of choosing an element s of a set S uniformly at random. $w \leftarrow \mathcal{A}(x, y, \ldots)$ represents the act of running the algorithm $\mathcal{A}$ with inputs $x, y, \ldots$ and producing output w. We write $w \leftarrow \mathcal{A}^{\mathcal{O}}(x, y, \ldots)$ for representing an algorithm $\mathcal{A}$ having access to an oracle $\mathcal{O}$. We denote by $\Pr[E]$ the probability that the event E occurs.

2.2 Public-Key Encryption Schemes

A Public Key Encryption Scheme (PKE) is defined as follows:

Definition 1 *(Public-Key Encryption). A public-key encryption scheme is a triplet of algorithms (*Gen, Enc, Dec*) such that:*

- Gen *is a probabilistic polynomial-time key generation algorithm which takes as input a security parameter* 1^k *and outputs a public key* PK *and a secret key* DK*. The public key specifies the message space* $\mathcal{M}$ *and the ciphertext space* $\mathcal{C}$*.*
- Enc *is a (possibly) probabilistic polynomial-time encryption algorithm which receives as input a public key* PK *and a message* $m \in \mathcal{M}$*, and outputs a ciphertext* $C \in \mathcal{C}$*.*
- Dec *is a deterministic polynomial-time decryption algorithm which takes as input a secret key* DK *and a ciphertext* C*, and outputs either a message* $m \in \mathcal{M}$ *or an error symbol* $\perp$*.*
- *(Soundness) For any pair of public and private keys generated by* Gen *and any message* $m \in \mathcal{M}$ *it holds that* $\mathsf{Dec}(DK, \mathsf{Enc}(PK, m)) = m$ *with overwhelming probability over the randomness used by* Gen *and* Enc*.*

Below we define indistinguishability against chosen-plaintext attacks (IND-CPA) [3] and against adaptive chosen-ciphertext attacks (IND-CCA2) [11]. Our game definition follows the approach of [5].

Definition 2 *(IND-CPA security). To a two-stage adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *against a PKE* Π *we associate the following experiment* $\mathsf{Exp}^{cpa}_{\Pi,\mathcal{A}}(k)$*:*

$(PK, DK) \leftarrow \mathsf{Gen}(1^k)$
$(m_0, m_1, state) \leftarrow \mathcal{A}_1(PK)$ *s.t.* $|m_0| = |m_1|$
$b \leftarrow \{0, 1\}$
$C^* \leftarrow \mathsf{Enc}(PK, m_b)$
$b' \leftarrow \mathcal{A}_2(C^*, state)$
If $b = b'$ *return* 1 *else return* 0

We define the advantage of $\mathcal{A}$ *in the experiment as*

$$\mathsf{Adv}^{cpa}_{\Pi,\mathcal{A}}(k) = |Pr[\mathsf{Exp}^{cpa}_{\Pi,\mathcal{A}}(k) = 1] - \tfrac{1}{2}|$$

We say that Π *is indistinguishable against chosen-plaintext attacks (IND-CPA) if for all probabilistic polynomial time (PPT) adversaries* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *the advantage of* $\mathcal{A}$ *in the experiment is a negligible function of* k*.*

Definition 3 *(IND-CCA2 security). To a two-stage adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *against a PKE* Π *we associate the following experiment* $\mathsf{Exp}^{cca2}_{\Pi,\mathcal{A}}(k)$*:*

$(PK, DK) \leftarrow \mathsf{Gen}(1^k)$
$(m_0, m_1, state) \leftarrow \mathcal{A}_1^{\mathsf{Dec}(DK,\cdot)}(PK)$ *s.t.* $|m_0| = |m_1|$
$b \leftarrow \{0, 1\}$
$C^* \leftarrow \mathsf{Enc}(PK, m_b)$
$b' \leftarrow \mathcal{A}_2^{\mathsf{Dec}(DK,\cdot)}(C^*, state)$
If $b = b'$ *return* 1 *else return* 0

The decryption oracle knows the secret key DK *and for each valid ciphertext submitted by the adversary as a query it sends the corresponding plaintext to the adversary. The adversary* $\mathcal{A}_2$ *is not allowed to query* $\mathsf{Dec}(DK, \cdot)$ *with* C^**. We define the advantage of* $\mathcal{A}$ *in the experiment as*

$$\mathsf{Adv}^{cca2}_{\Pi,\mathcal{A}}(k) = |Pr[\mathsf{Exp}^{cca2}_{\Pi,\mathcal{A}}(k) = 1] - \tfrac{1}{2}|$$

We say that Π *is indistinguishable against adaptive chosen-ciphertext attacks (IND-CCA2) if for all probabilistic polynomial time (PPT) adversaries* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *that makes a polynomial number of oracle queries the advantage of* $\mathcal{A}$ *in the experiment is a negligible function of* k*.*

2.3 Signature Schemes

Below we explain signature schemes and define one-time strong unforgeability.

Definition 4 *(Signature Scheme). A signature scheme is a triplet of algorithms (*Gen*,* Sig*,* Ver*) such that:*

- Gen *is a probabilistic polynomial-time key generation algorithm which takes as input a security parameter* 1^k *and outputs a verification key* vk *and a signing key* sk*. The verification key specifies the message space* $\mathcal{M}$ *and the signature space* $\mathcal{S}$*.*
- Sig *is a (possibly) probabilistic polynomial-time signing algorithm which receives as input a signing key* sk *and a message* $m \in \mathcal{M}$*, and outputs a signature* $\sigma \in \mathcal{S}$*.*
- Ver *is a deterministic polynomial-time verification algorithm which takes as input a verification key* vk*, a message* $m \in \mathcal{M}$ *and a signature* $\sigma \in \mathcal{S}$*, and outputs a bit indicating whether* σ *is a valid signature for* m *or not (i.e., the algorithm outputs* 1 *if it is a valid signature and outputs* 0 *otherwise).*
- *For any pair of signing and verification keys generated by* Gen *and any message* $m \in \mathcal{M}$ *it holds that* $\mathsf{Ver}(vk, m, \mathsf{Sig}(sk, m)) = 1$ *with overwhelming probability over the randomness used by* Gen *and* Sig*.*

Definition 5 *(One-Time Strong Unforgeability). To a two-stage adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *against a signature scheme* Σ *we associate the following experiment* $\mathsf{Exp}^{otsu}_{\Sigma,\mathcal{A}}(k)$*:*

$(vk, sk) \leftarrow \mathsf{Gen}(1^k)$
$(m, state) \leftarrow \mathcal{A}_1(vk)$
$\sigma \leftarrow \mathsf{Sig}(sk, m)$
$(m^*, \sigma^*) \leftarrow \mathcal{A}_2(m, \sigma, state)$
If $\mathsf{Ver}(vk, m^*, \sigma^*) = 1$ *and* $(m^*, \sigma^*) \neq (m, \sigma)$ *return* 1*, else return* 0

We say that a signature scheme Σ *is one-time strongly unforgeable if for all probabilist polynomial time (PPT) adversaries* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *the probability that* $\mathsf{Exp}^{otsu}_{\Sigma,\mathcal{A}}(k)$ *outputs* 1 *is a negligible function of* k*.*

2.4 Non-Interactive Zero-Knowledge

Now we define the notion of Adaptively-Secure Non-Interactive Zero-Knowledge. This definition is from [6].

Definition 6 *(Adaptively-Secure Non-Interactive Zero-Knowledge). A pair of probabilistic polynomial-time algorithms* $(\mathcal{P}, \mathcal{V})$ *is an adaptive, non-interactive zero-knowledge proof system for a language* $L \in \mathrm{NP}$ *if there exists a polynomial* poly *such that:*

Completeness: *For all* $x \in L \cap \{0,1\}^k$ *and all witnesses* w *for* x*,*

$$\Pr[r \leftarrow \{0,1\}^{\mathrm{poly}(k)}; \pi \leftarrow \mathcal{P}(r, x, w) : \mathcal{V}(r, x, \pi) = 1] = 1$$

Soundness: *For all (possibly unbounded) algorithms* $\mathcal{P}^*$*, the following is negligible in* k*:*

$$\Pr[r \leftarrow \{0,1\}^{\mathrm{poly}(k)}; (x, \pi) \leftarrow \mathcal{P}^*(r) : \mathcal{V}(r, x, \pi) = 1 \wedge x \in \{0,1\}^k \setminus L].$$

Zero-Knowledge: *Let* $(\mathcal{S}_1, \mathcal{S}_2)$ *and* $(\mathcal{A}_1, \mathcal{A}_2)$ *be a pair of two-staged algorithms. Consider the following experiments where* $x \in L \cap \{0,1\}^k$*:*

Game $\mathrm{ZK}_{\mathrm{real}}$	Game $\mathrm{ZK}_{\mathrm{sim}}$
$r \leftarrow \{0,1\}^{\mathrm{poly}(k)}$	$(r, state') \leftarrow \mathcal{S}_1(1^k)$
$(x, w, state) \leftarrow \mathcal{A}_1(r)$	$(x, w, state) \leftarrow \mathcal{A}_1(r)$
$\pi \leftarrow \mathcal{P}(r, x, w)$	$\pi \leftarrow \mathcal{S}_2(x, state')$
$b \leftarrow \mathcal{A}_2(r, x, \pi, state)$	$b \leftarrow \mathcal{A}_2(r, x, \pi, state)$

We require that there exist a probabilistic polynomial-time simulator $(\mathcal{S}_1, \mathcal{S}_2)$ *such that for any probabilistic polynomial-time algorithm* $(\mathcal{A}_1, \mathcal{A}_2)$ *the following is negligible in* k*:*

$$|\Pr_{\mathrm{ZK}_{\mathrm{real}}}[\mathcal{A}_2 \text{ outputs } 0] - \Pr_{\mathrm{ZK}_{\mathrm{sim}}}[\mathcal{A}_2 \text{ outputs } 0]|.$$

2.5 The DDN Cryptosystem

In this section, we briefly review the DDN cryptosystem. The DDN cryptosystem is constructed as follows. Let $\Pi = (\mathsf{CPA.Gen}, \mathsf{CPA.Enc}, \mathsf{CPA.Dec})$ be an IND-CPA secure PKE scheme and $\Sigma = (\mathsf{OT.Gen}, \mathsf{OT.Sig}, \mathsf{OT.Ver})$ be an one-time strongly unforgeable signature scheme. For simplicity, we assume that the size of signature of Σ is k where k is the security parameter. Then, we construct another PKE scheme as follows.

Key Generation. For security parameter k, run $\mathsf{CPA.Gen}(1^k)$ for $2k$ times to obtain

$$DK = (dk_{0,1}, ..., dk_{0,k}, dk_{1,1}, ..., dk_{1,k})$$

and

$$PK' = (pk_{0,1}, ..., pk_{0,k}, pk_{1,1}, ..., pk_{1,k}).$$

Pick a random string R (this will be used by the sender for generating a non-interactive zero-knowledge (NIZK) proof). The decryption key is DK and the public key is $PK = (PK', R)$.

Encryption. For encrypting m, run $\mathsf{OT.Gen}(1^k)$ to obtain a verification key vk and a signing key sk. Let vk_i be i-th bit of vk (assuming that vk is expressed as a k-bit binary string). Then, run $C_i \leftarrow \mathsf{CPA.Enc}(pk_{vk_i,i}, m)$ for $1 \le i \le k$. Generate a NIZK proof π which guarantees that the plaintexts of all C_i are identical.[3] Finally, run $\sigma \leftarrow \mathsf{OT.Sig}((C_1, ..., C_k, \pi), sk)$. The ciphertext is $C = (C_1, ..., C_k, \pi, vk, \sigma)$.

Decryption. For decrypting a given ciphertext $C = (C_1, ..., C_k, \pi, vk, \sigma)$, run $\mathsf{Ver}((C_1, ..., C_k, \pi), \sigma, vk)$ to check the validity of the signature. Also, test the validity of π. If invalid (for any of these tests), output $\perp$. Otherwise, run $m \leftarrow \mathsf{CPA.Dec}(C_1, dk_{vk_1,1})$.

3 Our Improved Construction

3.1 The Scheme

Here, we present our improved version of the DDN cryptosystem.

Key Generation. For security parameter k, run $\mathsf{CPA.Gen}(1^k)$ for $2k$ times to obtain

$$DK = (dk_{00,1}, ..., dk_{00,k/2}, dk_{01,1}, ..., dk_{01,k/2}, dk_{10,1}, ..., dk_{10,k/2}, dk_{11,1}, ..., dk_{11,k/2})$$

and

$$PK' = (pk_{00,1}, ..., pk_{00,k/2}, pk_{01,1}, ..., pk_{01,k/2}, pk_{10,1}, ..., pk_{10,k/2}, pk_{11,1}, ..., pk_{11,k/2}).$$

Pick a random string R. The decryption key is DK and the public key is $PK = (PK', R)$.

Encryption. For encrypting m, run $\mathsf{OT.Gen}(1^k)$ to obtain a verification key vk and a signing key sk. Let vk_i be i-th bit of vk (assuming that vk is expressed as a k-bit binary string). Then, run $C_i \leftarrow \mathsf{CPA.Enc}(pk_{vk_{2j-1}vk_{2j},j}, m)$ for $1 \le j \le k/2$. Generate a NIZK proof π which guarantees that the plaintexts of all C_i are identical.[4] Finally, run $\sigma \leftarrow \mathsf{OT.Sig}((C_1, ..., C_{k/2}, \pi), sk)$. The ciphertext is $C = (C_1, ..., C_{k/2}, \pi, vk, \sigma)$.

Decryption. For decrypting a given ciphertext $C = (C_1, ..., C_{k/2}, \pi, vk, \sigma)$, run $\mathsf{Ver}((C_1, ..., C_{k/2}, \pi), \sigma, vk)$ to check the validity of the signature. Also, test the validity of π. If invalid (for any of these tests), output $\perp$. Otherwise, run $m \leftarrow \mathsf{CPA.Dec}(C_1, dk_{vk_1vk_2,1})$.

3.2 Security Proof

Our scheme in the previous section is as secure as the original DDN scheme. Namely, it is IND-CCA2 secure assuming that Π is IND-CPA secure, Σ is one-time strongly unforgeable and that the NIZK proof system used is Adaptively-Secure.

Theorem 1 *The scheme presented in the previous section is an IND-CCA2 secure assuming that Π is an IND-CPA secure PKE, Σ is an one-time strongly unforgeable signature scheme and the Non-Interactive Zero-Knowledge proof system is Adaptively-Secure.*

---

[3] Since this is an NP language, NIZK proofs can be generated.

[4] Since this is an NP language, NIZK proofs can be generated.

Here, we give the proof of security of the proposed scheme. For proving the security, by using an algorithm $\mathcal{A}$ which breaks IND-CCA2 security of the proposed scheme, we construct another algorithm $\mathcal{B}$ which breaks IND-CPA security of a PKE scheme Π' which is as follows.

Key Generation. For security parameter k, run $\mathsf{CPA.Gen}(1^k)$ for $k/2$ times to obtain

$$\tilde{DK} = (dk_1, ..., dk_{k/2})$$

and

$$\tilde{PK} = (pk_1, ..., pk_{k/2}).$$

The decryption key is $\tilde{DK}$ and the public key is $\tilde{PK}$.

Encryption. For encrypting m, run $C_i \leftarrow \mathsf{CPA.Enc}(pk_j, m)$ for $1 \leq j \leq k/2$.

Decryption. For decrypting a given ciphertext $C = (C_1, ..., C_{k/2})$, run $m_j \leftarrow \mathsf{CPA.Dec}(C_j, dk_j)$ for $1 \leq j \leq k/2$. If $m_1 = ... = m_{k/2}$, output m_1. Otherwise, output $\perp$.

Due to the hybrid argument, it is clear that the above scheme Π' is IND-CPA secure if Π is IND-CPA secure.

For a given public key $\tilde{PK} = (pk_1, ..., pk_{k/2})$, $\mathcal{B}$ works as follows.

Key Generation. $\mathcal{B}$ generates public key $PK = (PK', R)$ which will be input to $\mathcal{A}$ where

$$PK' = (pk_{00,1}, ..., pk_{00,k/2}, pk_{01,1}, ..., pk_{01,k/2}, pk_{10,1}, ..., pk_{10,k/2}, pk_{11,1}, ..., pk_{11,k/2})$$

is generated as

1. Run $(vk^\star, sk^\star) \leftarrow \mathsf{OT.Gen}(1^k)$. Let $vk_i^\star$ be i-th bit of $vk^\star$.
2. For $1 \leq j \leq k/2$, for $b\beta \in \{00, 01, 10, 11\}$,
   (a) If $b\beta = vk^\star_{2j-1}vk^\star_{2j}$, then set $pk_{b\beta,j} = pk_j$.
   (b) If $b\beta \neq vk^\star_{2j-1}vk^\star_{2j}$, then run $(dk, pk) \leftarrow \mathsf{CPA.Gen}(1^k)$ and set $pk_{b\beta,j} = pk$. Also, set $dk_{b\beta,j} = dk$.

R is also appropriately generated by $\mathcal{B}$. $\mathcal{B}$ inputs PK to $\mathcal{A}$ as a public key of the proposed scheme.

Responding to Queries. When $\mathcal{A}$ submits a decryption query $C = (C_1, ..., C_{k/2}, \pi, vk, \sigma)$, $\mathcal{B}$ verifies the validity of C, and if invalid, outputs $\perp$. Otherwise, it selects j such that $vk_{2j-1}vk_{2j} \neq vk^\star_{2j-1}vk^\star_{2j}$ (if $vk \neq vk^\star$, there always exists such j), computes $m_j \leftarrow \mathsf{CPA.Dec}(C_j, dk_{vk_{2j-1}vk_{2j},j})$, and returns m_j to $\mathcal{A}$. If $vk = vk^\star$, $\mathcal{B}$ halts.

Constructing the Challenge Ciphertext. When $\mathcal{A}$ submits the two messages m_0 and m_1 which will be challenged, $\mathcal{B}$ also submits m_0 and m_1 to its own challenge oracle. Let $(C^\star_1, ..., C^\star_{k/2})$ be the challenge ciphertext for Π'. Then, $\mathcal{B}$ generates an NIZK proof $\pi^\star$ which guarantees that decryption results of $C^\star_1, ..., C^\star_{k/2}$ are identical (notice that since R is generated by $\mathcal{B}$, by using a certain trapdoor information $\mathcal{B}$ can generate such $\pi^\star$ without knowing the plaintext nor internal coin for encryption). Finally, $\mathcal{B}$ computes $\sigma^\star \leftarrow \mathsf{OT.Sig}((C^\star_1, ..., C^\star_{k/2}, \pi^\star), sk^\star)$, and sends $C^\star = (C^\star_1, ..., C^\star_{k/2}, \pi^\star, vk^\star, \sigma^\star)$ to $\mathcal{A}$ as the challenge ciphertext for the proposed scheme.

Output. When $\mathcal{A}$ outputs its guess b, $\mathcal{B}$ also outputs the same bit as its own guess.

Simulation fails only when $\mathcal{A}$ submits a valid query with $vk = vk^\star$ or when $\mathcal{A}$ produces a NIZK proof of validity of a ciphertext in which C_i and C_j (for some $i, j \in \{1, \ldots, k/2\}$) are ciphertexts of different plaintexts. However, the probability that the first event happens is negligible since Σ is strongly unforgeable and the probability that the second event happens is also negligible since the used NIZK proof system is Adaptively-Secure (and so meets the soundness requirement). Obviously, $\mathcal{B}$'s advantage is the same as $\mathcal{A}$'s (minus the probability that $\mathcal{A}$ produces one of the two events above).

Let Forge be the event that for some decryption query made by $\mathcal{A}$ we have that $\mathsf{Ver}((C_1, ..., C_{k/2}, \pi), \sigma, vk) = 1$ and $vk = vk^*$.

Lemma 1. $\Pr[\mathsf{Forge}]$ *is negligible.*

Proof. Assume that for a PPT adversary $\mathcal{A}$ against our scheme the forge probability ($\Pr[\mathsf{Forge}]$) is non-negligible, then we construct an adversary $\mathcal{B}'$ that forges a signature with the same probability. $\mathcal{B}'$ simulates the IND-CCA2 interaction for $\mathcal{A}$ as follows:

Key Generation: $\mathcal{B}'$ invokes the key generation algorithm of the signature scheme and obtains vk^*. It calls Π key generation algorithm $2k$ times obtaining the public keys

$$pk_{00,1}, ..., pk_{00,k/2}, pk_{01,1}, ..., pk_{01,k/2}, pk_{10,1}, ..., pk_{10,k/2}, pk_{11,1}, ..., pk_{11,k/2},$$

which constitute PK', and the secret keys

$$dk_{00,1}, ..., dk_{00,k/2}, dk_{01,1}, ..., dk_{01,k/2}, dk_{10,1}, ..., dk_{10,k/2}, dk_{11,1}, ..., dk_{11,k/2},$$

which constitute DK. R is also appropriately generated by $\mathcal{B}'$. $\mathcal{B}'$ inputs $PK = (PK', R)$ to $\mathcal{A}$ as a public key of the proposed scheme.

Decryption Queries: Whenever $\mathcal{A}$ makes a decryption query, $\mathcal{B}'$ proceeds as follows:

1. If for this ciphertext $vk = vk^*$ and $\mathsf{Ver}((C_1, ..., C_{k/2}, \pi), \sigma, vk) = 1$, $\mathcal{B}'$ outputs $((C_1, ..., C_{k/2}, \pi), \sigma)$ as the forgery and halts.
2. Otherwise, $\mathcal{B}'$ decrypts the ciphertext using the procedures of our scheme.

Challenging Query: Whenever $\mathcal{A}$ makes the challenging query with two messages $m_0, m_1 \in \mathcal{M}$ such that $|m_0| = |m_1|$, $\mathcal{B}'$ proceeds as follows:

1. Chooses randomly $b \in \{0, 1\}$.
2. Encrypts the message m_b using the procedures of our scheme. This is possible because $\mathcal{B}'$ can ask the signature oracle to sign one message, so it asks the oracle to sign the value $(C_1, ..., C_{k/2}, \pi)$ obtained during the encryption process.

As long as the event Forge did not occur, the simulation is perfect, so the probability that $\mathcal{B}'$ breaks the one-time strongly unforgeable signature scheme is exactly $\Pr[\mathsf{Forge}]$. Since the signature scheme is strongly unforgeable by assumption, $\Pr[\mathsf{Forge}]$ is negligible for all PPT adversaries against our scheme.

Let $\mathsf{V_{ILL}}$ be the event that $\mathcal{A}$ produces a NIZK proof of validity of a ciphertext in which C_i and C_j are ciphertexts of different plaintexts for some $i, j \in \{1, \dots, k/2\}$.

Lemma 2. $\Pr[\mathsf{V_{ILL}}]$ *is negligible.*

Proof. Assume that for a PPT adversary $\mathcal{A}$ against our scheme $\Pr[\mathsf{V_{ILL}}]$ is non-negligible, then we construct an adversary $\mathcal{B}''$ that violates the soundness of the NIZK proof system with the same probability. $\mathcal{B}''$ simulates the IND-CCA2 interaction for $\mathcal{A}$ as follows:

Key Generation: $\mathcal{B}''$ receives R as input. It calls Π key generation algorithm $2k$ times obtaining the public keys

$$pk_{00,1}, ..., pk_{00,k/2}, pk_{01,1}, ..., pk_{01,k/2}, pk_{10,1}, ..., pk_{10,k/2}, pk_{11,1}, ..., pk_{11,k/2},$$

which constitute PK', and the secret keys

$$dk_{00,1}, ..., dk_{00,k/2}, dk_{01,1}, ..., dk_{01,k/2}, dk_{10,1}, ..., dk_{10,k/2}, dk_{11,1}, ..., dk_{11,k/2},$$

which constitute DK. $\mathcal{B}''$ inputs $PK = (PK', R)$ to $\mathcal{A}$ as a public key of the proposed scheme.

Decryption Queries: Whenever $\mathcal{A}$ makes a decryption query, $\mathcal{B}''$ proceeds as follows:

1. If for this ciphertext π is valid and C_i and C_j are ciphertexts of different plaintexts for some $i, j \in \{1, \dots, k/2\}$, $\mathcal{B}''$ outputs $((C_1, ..., C_{k/2}), \pi)$ and halts.
2. Otherwise, $\mathcal{B}''$ decrypts the ciphertext using the procedures of our scheme.

Challenging Query: Whenever $\mathcal{A}$ makes the challenging query with two messages $m_0, m_1 \in \mathcal{M}$ such that $|m_0| = |m_1|$, $\mathcal{B}''$ proceeds as follows:

1. Chooses randomly $b \in \{0, 1\}$.
2. Encrypts the message m_b using the procedures of our scheme.

As long as the event $\mathsf{V_{ILL}}$ did not occur, the simulation is perfect, so the probability that $\mathcal{B}''$ breaks the soundness of the NIZK proof system is exactly $\Pr[\mathsf{V_{ILL}}]$. Since the NIZK proof system is Adaptively-Secure by assumption, $\Pr[\mathsf{V_{ILL}}]$ is negligible for all PPT adversaries against our scheme.

3.3 Performance: Comparison with the Original DDN

The most remarkable advantage of our construction to the original DDN is that its ciphertext length and computational cost are significantly reduced *without sacrificing anything*. More specifically, in our scheme the number of component ciphertexts (i.e. $(C_1, ..., C_{k/2})$) is reduced to be a half of that of the original scheme (i.e. $(C_1, ..., C_k)$) without increasing the key size. Furthermore, since the NIZK proof for guaranteeing that decryption results of $C_1, ..., C_{k/2}$ are identical can also be significantly simpler than that of $C_1, ..., C_k$, the component for the NIZK proof can also be reduced. In summary, performance of our construction is equivalent or superior to that of the original DDN *in all aspects*. We summarize below the properties of our construction in comparison to the original DDN.

Ciphertext Size	nearly half
Public Key Size	same
Secret Key Size	same
Encryption Computational Cost	nearly half
Decryption Computational Cost	nearly same

4 An Intuitive Explanation of Our Trick

Here, we give a high level overview of our trick for the improvement. For explaining it, we start with the following observation. By carefully looking into the security proof of the original DDN, we notice that it may be still valid and actually works for the following generalized construction:

Key Generation. For security parameter k, run $\mathsf{CPA.Gen}(1^k)$ for nN times to obtain

$$DK = (dk_{1,1}, ..., dk_{1,N}, dk_{2,1}, ..., dk_{2,N}, ..., dk_{n,1}..., dk_{n,N})$$

and

$$PK' = (pk_{1,1}, ..., pk_{1,N}, pk_{2,1}, ..., pk_{2,N}, ..., pk_{n,1}..., pk_{n,N}).$$

Pick a random string R. The decryption key is DK and the public key is $PK = (PK', R)$.

Encryption. For encrypting m, run $\mathsf{OT.Gen}(1^k)$ to obtain a verification key vk and a signing key sk. Assume that vk can be expressed as $(vk_1, vk_2, ..., vk_N) \in \{1, ..., n\}^N$ such that for all $i \in \{1, ..., N\}$, $vk_i \in \{1, ..., n\}$. Then, run $C_i \leftarrow \mathsf{CPA.Enc}(pk_{vk_j,j}, m)$ for $1 \leq j \leq N$. Generate a NIZK proof π which guarantees that the plaintexts of all C_i are identical. Finally, run $\sigma \leftarrow \mathsf{OT.Sig}((C_1, ..., C_N, \pi), sk)$. The ciphertext is $C = (C_1, ..., C_N, \pi, vk, \sigma)$.

Decryption. For decrypting a given ciphertext $C = (C_1, ..., C_N, \pi, vk, \sigma)$, run $\mathsf{Ver}((C_1, ..., C_N, \pi), \sigma, vk)$ to check the validity of the signature. Also, test the validity of π. If invalid (for any of these tests), output $\perp$. Otherwise, run $m \leftarrow \mathsf{CPA.Dec}(C_1, dk_{vk_1,1})$.

The above scheme is provably IND-CCA2 secure if $n^N \geq 2^k$. The proof can be trivially obtained from that presented in the previous sections of this paper.

From this observation, we see that there asymptotically exists a trade-off between N and $n \times N$, and this implies that if we reduce the ciphertext size (in other words, decrease N), then we have to increase the key size (in other words, increase $n \times N$). Let $|KEY|$ and $|CTXT|$ be nN and N, respectively. Then, we have the following inequality:

$$|KEY| \geq |CTXT| \cdot 2^{k|CTXT|^{-1}}.$$

This inequality implies that if $|CTXT|$ decreases, then $|KEY|$ *exponentially* increases. Hence, the above generalized DDN is not very useful in a general sense. However, this is an *asymptotic observation* and does not always hold. Especially, the righthand side of the above inequality is not monotonic in $|CTXT|$ (it has a local minimum at $|CTXT| = k/(\log_2 e)$), and therefore, there exist some interesting parameter settings for $k/2 \leq |CTXT| \leq k$

Based on this observation, we can construct some interesting variants of the DDN as well. For example, if we set $|CTXT| = k/2$, then $|KEY|$ becomes the same value as that for $|CTXT| = k$. This means that we can compress the ciphertext length without increasing the key size. (This example is what we presented in the previous sections). Another interesting example is that by setting $|CTXT| \simeq k/(\log_2 e)$, we can simultaneously reduce both $|CTXT|$ and $|KEY|$. Many other interesting variations seem also be possible.

5 Conclusions

We have modified the DDN cryptosystem and obtained a cryptosystem that reduces almost by half the ciphertext size without changing either the cryptosystem security or its keys sizes. Our improvement is based on a trick that asymptotically allows a trade-off between the ciphertext size and the keys sizes, but for our specific parameters allows a reduction of the ciphertext size without changing the other parameters of the cryptosystem.

References

1. D. Dolev, C. Dwork, M. Naor. Non-malleable Cryptography. SIAM J. Comput. 30(2): 391-437 (2000).
2. R. Dowsley and J. Müller-Quade and A. C. A. Nascimento, A CCA2 Secure Public Key Encryption Scheme Based on the McEliece Assumptions in the Standard Model, to appear , CT-RSA 2009, Also available from http://eprint.iacr.org/2008/468
3. S. Goldwasser, S. Micali: Probabilistic Encryption. J. Comput. Syst. Sci. 28(2): 270-299 (1984).
4. G. Hanaoka, H. Imai, K. Ogawa, H. Watanabe: Chosen Ciphertext Secure Public Key Encryption with a Simple Structure. IWSEC 2008: 20-33
5. D. Hofheinz, E. Kiltz. Secure Hybrid Encryption from Weakened Key Encapsulation. CRYPTO 2007: 553-571.
6. J. Katz. Lecture Notes. Available at http://www.cs.umd.edu/ jkatz/gradcrypto2/scribes.html.
7. Y. Lindell. A Simpler Construction of CCA2-Secure Public-Key Encryption under General Assumptions. *EUROCRYPT 2003*. pp. 241-254. 2003.
8. M. Naor and M. Yung. Universal One-Way Hash Functions and their Cryptographic Applications. In 21st STOC, pages 3343, 1989.
9. R. Pass, A. Shelat, V. Vaikuntanathan: Construction of a Non-malleable Encryption Scheme from Any Semantically Secure One. CRYPTO 2006: 271-289
10. C. Peikert, B. Waters. Lossy trapdoor functions and their applications. *STOC 2008*. pp. 187-196. 2008.
11. C. Rackoff, D. R. Simon: Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. CRYPTO 1991: 433-444.
12. A. Rosen and G. Segev. Chosen-Ciphertext Security via Correlated Products. Available at http://eprint.iacr.org/2008/116. 2008.
13. A. Sahai. Non-Malleable Non-Interactive Zero Knowledge and Adaptive Chosen- Ciphertext Security. In *40th FOCS*, pages 543553, 1999.