Plaintext-Checkable Encryption

Sébastien Canard, Georg Fuchsbauer, Aline Gouget, Fabien Laguillaumie

Plaintext-Checkable Encryption

Sébastien Canard[1], Georg Fuchsbauer[2], Aline Gouget[3], and Fabien Laguillaumie[4]

[1] Orange Labs, Applied Crypto Group, Caen, France
[2] University of Bristol, Dept. Computer Science, UK
[3] Gemalto, Security Lab, Meudon, France
[4] UCBN and CNRS/ENSL/INRIA/UCBL LIP, Lyon, France

Abstract. We study the problem of searching on encrypted data, where the search is performed using a plaintext message or a keyword, rather than a message-specific trapdoor as done by state-of-the-art schemes. The use cases include delegation of key-word search e.g. to a cloud data storage provider or to an email server, using a plaintext message. We define a new cryptographic primitive called *plaintext-checkable encryption* (PCE), which extends public-key encryption by the following functionality: given a plaintext, a ciphertext and a public key, it is universally possible to check whether the ciphertext encrypts the plaintext under the key. We provide efficient generic random-oracle constructions for PCE based on any probabilistic or deterministic encryption scheme; we also give a practical construction in the standard model. As another application we show how PCE can be used to improve the efficiency in group signatures with *verifier-local revocation* (VLR) and backward unlinkability. These group signatures provide efficient revocation of group members, which is a key issue in practical applications.

Keywords. Deterministic/probabilistic encryption, unlinkability, group signature with VLR and backward unlinkability.

1 Introduction

The problem of searching on data that is encrypted has been studied intensively and in many different scenarios. For instance, the problem of delegation of keyword search on private databases to a data storage provider concerns users who upload their data to a provider they do not fully trust. When the user wants to delegate keyword search on his own encrypted data to the provider, he usually has to transmit a corresponding message-dependent trapdoor (or encrypted keyword) which enables the provider to perform the search. When the databases are public, the user wishes to delegate the search on public data to a cloud data storage provider without revealing the plaintext content of the search. Another setting is the delegation of search to an email gateway [9], where data collected by the mail server is from third parties (contrary to the private-key setting as above) and the database is not public.

Most of the constructions proposed in the literature are based either on symmetric-key cryptography to encrypt the plaintext message or keyword, or

Appeared in Orr Dunkelman (Ed.): CT-RSA 2012, volume 7178 of LNCS, pp. 332-348, 2012.

on searchable encryption without the ability to decrypt the message as done in [22]. The security of the search process in the state of the art of public-key encryption constructions has always been studied assuming that the search process uses a secret trapdoor and not a plaintext message. In this work we focus on this latter case, which is naturally related to public-key cryptography. This case can in practice be very useful when the database contains relations between different words (a name and a status for example) and it is these relations that have to be kept secret rather than the words themselves. Thus, when searching e.g. the number of persons having the status "important illness", the keyword "important illness" is not secret and can be directly used to perform the search. Many functionalities extending the basic setting of public-key encryption have been considered, in particular related to data search. For example, *decryptable searchable encryption* [13] allows someone having a trapdoor corresponding to a message, to test whether a given ciphertext encrypts this message. Another example is *encryption with equality test*, proposed in [23]. Using the equality test, one can check whether two ciphertexts encrypt the same plaintext.

In this paper we propose and study a new cryptographic primitive we call *plaintext-checkable encryption* (PCE). A plaintext-checkable encryption scheme is a probabilistic public-key encryption scheme with the additional functionality that anyone can test whether a ciphertext c is the encryption of a given plaintext message m under a public encryption key pk. Despite this functionality, we demand that the ciphertext leak as little information as possible about the plaintext. Of course, a PCE scheme cannot achieve the standard notion of *indistinguishability under chosen-plaintext attack*, as an adversary choosing two messages and receiving the encryption of one of them can simply test which message was encrypted. The same holds when the encryption algorithm is *deterministic*: an adversary can just re-encrypt candidate messages and thus break classical indistinguishability.

As was done in the case of deterministic encryption [3], we assume that the plaintexts are drawn from a space of large min-entropy; indistinguishability means thus the impossibility of distinguishing ciphertexts of messages drawn from different high min-entropy spaces. We show however that we can achieve a strictly stronger security notion than indistinguishability for deterministic encryption [3, 4]: an adversary is not able to distinguish two encryptions of the same message from encryptions of different messages. This notion cannot be achieved by deterministic encryption, since there is only one possible ciphertext per message, and encryption with equality check cannot achieve it either. We say that an encryption scheme satisfies *unlinkability* if no polynomial-time adversary can win the following game: a challenger draws two messages from a high min-entropy space of the adversary's choice and gives the adversary either encryptions of the two messages or two encryptions of one message, and the adversary has to decide which is the case. We relate this notion to the different types of indistinguishability, showing e.g. that it is strictly stronger than the indistinguishability notion for deterministic encryption, and we argue that our notion is sufficient for our applications. We provide efficient generic constructions

Appeared in Orr Dunkelman (Ed.): CT-RSA 2012, volume 7178 of LNCS, pp. 332-348, 2012.

of PCE schemes satisfying unlinkability based either on probabilistic or deterministic encryption with a security proof in the random-oracle model (ROM).[1] We also build a practical construction based on ElGamal encryption, secure in the standard model.

Apart from its immediate applications to searching on encrypted data, PCE lends itself naturally to improve the efficiency of group signatures with *verifier-local revocation* (VLR). Group signatures allow members of a group to sign on behalf of the group without revealing their individual identity. Group signatures with VLR were introduced by Boneh and Shacham [10] and allow efficient revocation of group members, which is a key issue in practical applications. In VLR group signatures the revocation messages only have to be sent to signature verifiers, as opposed to both signers and verifiers in previous schemes. We note that unlinkability of ciphertexts is precisely the property required by the encryptions contained in group signatures [8, 11]. We show that PCE can be used to encrypt a user-specific revocation token, like a certificate, which will be part of a group signature. A group member can then be revoked by publishing the token, as every verifier can apply the plaintext check to the encrypted token in order to determine whether it corresponds to a revoked user. Since tokens will be drawn from a high min-entropy space, two group signatures containing the same token are unlinkable by the security of the PCE. Our VLR group signature scheme achieves *backward unlinkability* and is proven secure in the standard model.

The paper is organized as follows. In Sect. 2 we formally define plaintext-checkable encryption and we give security definitions and compare them to existing security notions for public-key encryption. In Sect. 3 we provide generic constructions of PCE in the random-oracle model based on either deterministic or probabilistic encryption, while Sect. 4 gives the description of our practical construction in the standard model. We finally show in Sect. 5 how PCE can be used to design very practical group signatures with VLR. Due to space limitations, proofs are omitted but are available in the full version.

2 Plaintext-Checkable Encryption

We define here the notion of plaintext-checkable encryption and its security.

2.1 Definition of Plaintext-Checkable Encryption

Let $k \in \mathbb{N}$ be a security parameter. A *plaintext-checkable encryption scheme* (PCE for short) is composed of the following algorithms (of which the first 3 constitute a public-key encryption scheme).

- KeyGen is a probabilistic algorithm which takes as input 1^k and outputs a key pair (pk, sk) of public and secret key, respectively.

[1] It may be possible to design PCE schemes from any decryptable searchable encryption scheme by simply publishing trapdoors (one trapdoor per message or, in some cases, the master trapdoor). However, our constructions are more efficient.

Appeared in Orr Dunkelman (Ed.): CT-RSA 2012, volume 7178 of LNCS, pp. 332-348, 2012.

– $\mathsf{Encrypt}$ is a probabilistic algorithm which takes as inputs 1^k, a public key pk and a plaintext $m \in \{0,1\}^*$ and outputs a ciphertext c.
– $\mathsf{Decrypt}$ is a deterministic algorithm which takes as inputs 1^k, a ciphertext c and a secret key sk and outputs either a plaintext m or $\perp$.
– PCheck is a deterministic algorithm which takes as inputs 1^k, a ciphertext c, a public key pk and a putative message m. It outputs 1 if c is an encryption of m, and 0 otherwise.

These algorithms must verify the following properties of *correctness*.

Correctness of decryption: $\forall k \in \mathbb{N}$ and $m \in \{0,1\}^*$,
$\Pr\left[(pk, sk) \xleftarrow{\$} \mathsf{KeyGen}(1^k), c \xleftarrow{\$} \mathsf{Encrypt}(1^k, pk, m) : \mathsf{Decrypt}(1^k, sk, c) = m\right] = 1.$

Correctness of plaintext check (perfect consistency): $\forall k \in \mathbb{N}$ and $m \in \{0,1\}^*$,

$$\Pr\left[(pk, sk) \xleftarrow{\$} \mathsf{KeyGen}(1^k), c \xleftarrow{\$} \mathsf{Encrypt}(1^k, pk, m) : \mathsf{PCheck}(1^k, c, pk, m) = 1\right] = 1.$$

The property of perfect consistency is implied by the correctness of decryption and the two following properties, which guarantee that PCheck behaves as expected. The following two notions state that if a ciphertext decrypts to a plaintext then PCheck matches them (completeness) and if PCheck matches a ciphertext to a plaintext then the former encrypts the latter (soundness).

Checking completeness: no adversary is able to output a ciphertext c which decrypts to a message that is refused by PCheck on input c. Formally, for every $k \in \mathbb{N}$ and every probabilistic polynomial-time (p.p.t.) algorithm $\mathcal{A}$ that, on inputs 1^k and a public key pk, outputs a ciphertext c, the following probability should be negligible:

$$\Pr\left[(pk, sk) \xleftarrow{\$} \mathsf{KeyGen}(1^k), c \xleftarrow{\$} \mathcal{A}(1^k, pk), m \xleftarrow{\$} \mathsf{Decrypt}(1^k, c, sk) \ : \ \mathsf{PCheck}(1^k, pk, c, m) = 0\right].$$

Checking soundness: this property states that no adversary should be able to produce a plaintext and ciphertext such that the decryption and the check procedures do not agree on the plaintext related to c. More formally, for every $k \in \mathbb{N}$ and every p.p.t. algorithm $\mathcal{A}$ that, on inputs 1^k and a public key pk, outputs a ciphertext c and a plaintext $\tilde{m}$, the following probability should be negligible:

$$\Pr\left[(pk, sk) \xleftarrow{\$} \mathsf{KeyGen}(1^k), (c, \tilde{m}) \xleftarrow{\$} \mathcal{A}(1^k, pk), m \xleftarrow{\$} \mathsf{Decrypt}(1^k, c, sk) \ : \ m \neq \tilde{m} \wedge \mathsf{PCheck}(1^k, pk, c, \tilde{m}) = 1\right].$$

2.2 A Taxonomy of Indistinguishability

The classical property of indistinguishability (for public-key encryption schemes) cannot be achieved by a PCE due to the ability to check the plaintext messages

Appeared in Orr Dunkelman (Ed.): CT-RSA 2012, volume 7178 of LNCS, pp. 332-348, 2012.

$\underline{\mathbf{Exp}^{\mathsf{ind\text{-}cpa}}_{\Pi,\mathcal{A}}(k)}$

$b \xleftarrow{\$} \{0,1\}$
$(pk, sk) \leftarrow \mathcal{G}(1^k)$
$(m_0, m_1, st) \leftarrow \mathcal{A}_f(1^k, pk)$
$c \leftarrow \mathcal{E}(1^k, pk, m_b)$
$b' \leftarrow \mathcal{A}_g(1^k, c, st)$
Return $(b' = b)$

$\underline{\mathbf{Exp}^{\mathsf{unlink}}_{\Pi,\mathcal{A}}(k)}$

$b \xleftarrow{\$} \{0,1\}$
$(pk, sk) \leftarrow \mathcal{G}(1^k)$
$m_0 \leftarrow \mathcal{A}_f(1^k, pk)$
$m_1 \leftarrow \mathcal{A}_f(1^k, pk)$
$c_0 \leftarrow \mathcal{E}(1^k, pk, m_b)$
$c_1 \leftarrow \mathcal{E}(1^k, pk, m_1)$
$b' \leftarrow \mathcal{A}_g(1^k, pk, c_0, c_1)$
Return $(b' = b)$

$\underline{\mathbf{Exp}^{\mathsf{ind\text{-}det}}_{\Pi,\mathcal{A}}(k)}$

$b \xleftarrow{\$} \{0,1\}$
$m \leftarrow \mathcal{A}_f(1^k, b)$
$(pk, sk) \leftarrow \mathcal{G}(1^k)$
$c \leftarrow \mathcal{E}(1^k, pk, m)$
$b' \leftarrow \mathcal{A}_g(1^k, pk, c)$
Return $(b' = b)$

Fig. 1. Security experiments for indistinguishability of Π

(see below). We discuss in this section the properties of indistinguishability for encryption schemes.

In the following, we denote by $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ a secure encryption scheme. Depending on the context, Π can be either probabilistic (denoted Π_p) or deterministic (denoted Π_d). We first remark that a PCE can also be represented as an encryption scheme $(\mathcal{G}, \mathcal{E}, \mathcal{D}) = (\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$, in the notation from Sect. 2.1.

An adversary $\mathcal{A}$ is defined by a pair of algorithms denoted by $\mathcal{A} = (\mathcal{A}_f, \mathcal{A}_g)$, representing the find and guess stage of the experiment, respectively. The adversary $\mathcal{A}$ is said to be polynomial if each constituent algorithm has a running time polynomial in its input length. It is assumed that $\mathcal{A}_f$ and $\mathcal{A}_g$ share neither coins nor state. We study three security experiments for the indistinguishability properties of an encryption scheme Π; the three security experiments, denoted by $\mathbf{Exp}^{\mathsf{ind\text{-}cpa}}_{\Pi,\mathcal{A}}(k)$, $\mathbf{Exp}^{\mathsf{unlink}}_{\Pi,\mathcal{A}}(k)$ and $\mathbf{Exp}^{\mathsf{ind\text{-}det}}_{\Pi,\mathcal{A}}(k)$, are described in Fig. 1. We first define two classes of adversaries.

Definition 1 (High min-entropy). *An adversary $\mathcal{A} = (\mathcal{A}_f, \mathcal{A}_g)$ is* legitimate *if there exists a function $\ell(\cdot)$ s.t. for all c and all $m \in [\mathcal{A}_f(1^k, c)]$ we have $|m| = \ell(k)$ (where c can be a bit, as for ind-det adversaries, or a public key, as for ind-cpa and unlink adversaries).*
Moreover, we say that an adversary $\mathcal{A} = (\mathcal{A}_f, A_g)$ has min-entropy μ *if*

$$\forall k \in \mathbb{N}\ \forall c\ \forall m : \Pr\left[m' \leftarrow \mathcal{A}_f(1^k, b)\ :\ m' = m\right]\ \leq\ 2^{-\mu(k)}\ .$$

$\mathcal{A}$ is said to have high min-entropy *if it has min-entropy μ with $\mu(k) \in \omega(\log k)$.*

The first experiment $\mathbf{Exp}^{\mathsf{ind\text{-}cpa}}_{\Pi,\mathcal{A}}(k)$ represents the standard indistinguishability property for probabilistic encryption schemes.

Definition 2 (IND-CPA). *Let $k \in \mathbb{N}$, let $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be an encryption scheme, let $\boldsymbol{Exp}^{\mathsf{ind\text{-}cpa}}_{\Pi,\mathcal{A}}$ be as defined in Fig. 1 and denote $\boldsymbol{Adv}^{\mathsf{ind\text{-}cpa}}_{\Pi,\mathcal{A}}(k) := 2 \cdot \Pr\left[\boldsymbol{Exp}^{\mathsf{ind\text{-}cpa}}_{\Pi,\mathcal{A}}(k) \rightarrow \mathsf{true}\right] - 1$. We say that Π* satisfies indistinguishability under a chosen-plaintext attack *if for every legitimate p.p.t. adversary $\mathcal{A} = (\mathcal{A}_f, \mathcal{A}_g)$, the advantage $\boldsymbol{Adv}^{\mathsf{ind\text{-}det}}_{\Pi,\mathcal{A}}(k)$ is negligible.*

Appeared in Orr Dunkelman (Ed.): CT-RSA 2012, volume 7178 of LNCS, pp. 332-348, 2012.

The experiment $\mathbf{Exp}^{\mathsf{ind\text{-}det}}_{\Pi,\mathcal{A}}(k)$ is a simplified definition of the indistinguishability property for deterministic encryption introduced in [4], which has been shown to be equivalent to the original definition considered in [3]. We simplify the original definition by considering adversaries that produce distributions of *messages* rather than distributions of message vectors[2].

Definition 3 (IND-DET [4]). *Let $k \in \mathbb{N}$, let $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be an encryption scheme, and let $\boldsymbol{Exp}^{\mathsf{in\text{-}det}}_{\Pi,\mathcal{A}}$ be as defined in Fig. 1. Let $\boldsymbol{Adv}^{\mathsf{ind\text{-}det}}_{\Pi,\mathcal{A}}(k) := 2 \cdot \Pr\left[\boldsymbol{Exp}^{\mathsf{ind\text{-}det}}_{\Pi,\mathcal{A}}(k) \to \mathsf{true}\right] - 1$. We say Π* satisfies ind-det *if for every legitimate p.p.t. adversary $\mathcal{A} = (\mathcal{A}_f, \mathcal{A}_g)$ with high min-entropy, $\boldsymbol{Adv}^{\mathsf{ind\text{-}det}}_{\Pi,\mathcal{A}}(k)$ is negligible.*

We define the third security experiment as the infeasibility of deciding whether two ciphertexts encrypt the same message. The definition shares with ind-det that the messages have to be chosen from a high min-entropy space: otherwise the notion is not satisfiable by a plaintext-checkable scheme, since the adversary could simply check all messages. As we will show all along this paper, this security definition is achievable by plaintext-checkable schemes and sufficient for our applications.

Definition 4 (UNLINK). *Let $k \in \mathbb{N}$ and $\boldsymbol{Adv}^{\mathsf{unlink}}_{\Pi,\mathcal{A}}(k) := 2 \cdot \Pr\left[\boldsymbol{Exp}^{\mathsf{unlink}}_{\Pi,\mathcal{A}}(k) \to \mathsf{true}\right] - 1$, for an encryption scheme $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ with $\boldsymbol{Exp}^{\mathsf{unlink}}_{\Pi,\mathcal{A}}$ as defined in Fig. 1. We say Π* has unlinkable encryptions *(or "satisfies unlink") if for every legitimate p.p.t. adversary $\mathcal{A}$ with high min-entropy, $\boldsymbol{Adv}^{\mathsf{unlink}}_{\Pi,\mathcal{A}}(k)$ is negligible.*

We now give a complete taxonomy of all these security notions and we prove (see the full version) that the unlink notion falls strictly between ind-cpa of probabilistic encryption, and ind-det of deterministic encryption. More precisely, we show the following relation:

$$\textbf{IND-CPA} \subsetneq \textbf{UNLINK} \subsetneq \textbf{IND-DET}.$$

This means that every scheme that achieves ind-cpa is unlink and every scheme that achieves unlink is ind-det. On the other hand, there are schemes that are unlink but not ind-cpa, and others satisfying ind-det but not unlink.

It is obvious that a PCE scheme cannot be ind-cpa since the adversary could forward m_0 and m_1 as st from $\mathcal{A}_f$ to $\mathcal{A}_g$, which could then apply PCheck to the challenge c and for example m_0, and win the experiment with overwhelming probability. As a consequence, the somewhat best we can hope for in the case of PCE is unlinkability. We will thus show that our schemes satisfy this new security notion.

Deterministic encryption schemes [3], though trivially plaintext-checkable, cannot satisfy the property unlink since every two encryptions of a message are equal, which allows a trivial check of plaintext equality. One may attempt to construct plaintext-checkable encryption from an encryption scheme with equality

[2] The original definition considers vectors of messages since (unlike for ind-cpa-secure encryption) there is no reduction to the single-message case by a hybrid argument for deterministic encryption.

Appeared in Orr Dunkelman (Ed.): CT-RSA 2012, volume 7178 of LNCS, pp. 332-348, 2012.

test as described in [23] by simply encrypting the message and then performing the test of equality. However, this scheme does not satisfy unlink either for obvious reasons. Moreover, as noticed by the authors, their Test function only works properly when the ciphertexts are two real encryptions of messages, as this procedure does not check the validity of the ciphertexts.

It thus remains open to give a construction (practical or generic) with the above features, namely providing a PCheck procedure, while maintaining unlinkability. We give such constructions in the two following sections.

3 Generic Constructions for PCE in the ROM

We show how to obtain secure PCE schemes using a secure probabilistic or deterministic encryption scheme with security proofs in the random-oracle model.

3.1 A PCE based on a Probabilistic Encryption Scheme

In this construction a message m is encrypted by first choosing a random string r and computing a hash value ρ of the message and r. This value ρ is then used as the random coins of the probabilistic encryption algorithm to encrypt m, and r is added to the ciphertext. The algorithm PCheck consists essentially in re-computing ρ and then re-encrypting the message with random coins ρ and comparing it to the candidate ciphertext. Our solution is described in Fig. 2. The triple $\Pi_p = (\mathcal{G}_p, \mathcal{E}_p, \mathcal{D}_p)$ denotes a probabilistic encryption scheme satisfying indistinguishability under chosen-message attack, and $\mathcal{H} : \{0,1\}^* \to \{0,1\}^{\ell(k)}$ denotes a hash function modeled as a random oracle.

Algorithm $\mathsf{KeyGen}(1^k)$
$(\overline{pk}, \overline{sk}) \xleftarrow{\$} \Pi_p.\mathcal{G}_p(1^k)$
$pk \leftarrow \overline{pk}$
$sk \leftarrow \overline{sk}$
return (pk, sk)

Algorithm $\mathsf{Encrypt}(1^k, pk, m)$
$\overline{pk} \leftarrow pk$
$r \xleftarrow{\$} \{0,1\}^{\ell(k)}$
$\rho \leftarrow \mathcal{H}(m\|r)$
$\overline{c} \leftarrow \Pi_p.\mathcal{E}_p(1^k, \overline{pk}, m; \rho)$
$C \leftarrow (\overline{c}, r)$
return C

Algorithm $\mathsf{Decrypt}(1^k, sk, C)$
$(\overline{c}, r) \leftarrow C$
$\overline{sk} \leftarrow sk$
$m \leftarrow \Pi_p.\mathcal{D}_p(1^k, \overline{sk}, \overline{c})$
return m

Algorithm $\mathsf{PCheck}(1^k, pk, C, m)$
$(\overline{c}, r) \leftarrow C$
$\overline{pk} \leftarrow pk$
$\rho \leftarrow \mathcal{H}(m\|r)$
$\tilde{c} \leftarrow \Pi_p.\mathcal{E}_p(1^k, \overline{pk}, m; \rho)$
if $\tilde{c} = \overline{c}$ then return 1
else return 0

Fig. 2. Unlinkable PCE from an ind-cpa encryption scheme Π_p

Appeared in Orr Dunkelman (Ed.): CT-RSA 2012, volume 7178 of LNCS, pp. 332-348, 2012.

The following theorem states the security of the construction of Fig. 2, i.e. that it satisfies unlinkability. Essentially, the unlinkability of this construction follows from the indistinguishability of the underlying encryption scheme. However, quite some care needs to be taken to ensure that the simulation in the reduction is perfect, as the adversary against unlinkability may make queries to the random oracle that the simulator cannot answer.

Theorem 1. *If Π_p satisfies* ind-cpa *then the PCE from Fig. 2 satisfies* unlink.

Proof (sketch, see full version for the full proof). We show that a successful adversary $\mathcal{A}$ against unlink of our PCE scheme can be used to construct an adversary $\mathcal{B}$ against ind-cpa of Π_p. A natural construction of $\mathcal{B}$ is the following: $\mathcal{B}_f$ runs $\mathcal{A}_f$ twice and outputs the obtained messages m_0 and m_1. The challenger then gives $\mathcal{B}_g$ a Π_p-encryption c of m_b. Now $\mathcal{B}_g$ must use $\mathcal{A}_g$ to determine b. Playing the unlinkability game, $\mathcal{A}_g$ expects two PCE ciphertexts; one of m_b and one of m_1. While the latter can be computed honestly, $\mathcal{B}_g$ could construct the former as (c, r_0), for some random r_0.

However, this implicitly defines $\mathcal{H}(m_b, r_0)$ to be the randomness $\mathcal{B}$'s challenger used in constructing c; $\mathcal{B}$ can thus not answer this random-oracle query and the simulation might fail. In a series of lemmas, we show that under ind-cpa of Π_p, the probability of $\mathcal{A}_g$ (who does not know m_0 and m_1) querying m_0 or m_1 to $\mathcal{H}$ is negligible. We first show that this holds if $\mathcal{B}$'s challenger's bit $b = 0$:

Suppose in game unlink when $b = 0$, $\mathcal{A}_g$ queries $(m_0 \| r)$ (for some r) to the random oracle $\mathcal{H}$. Then we construct $\mathcal{B}'$ that breaks ind-cpa. It uses $\mathcal{A}_f$ to sample m_0 and m_1, gets an encryption c of m_d from its challenger and then runs $\mathcal{A}_g$ on (c, r_0) (for some random r_0) and a PCE encryption of an independent message m'. Since $\mathcal{A}_g$ does not have any information on m_{1-d} (which was sampled from a high min-entropy space), querying e.g. m_0 must mean $d = 0$. Thus if $\mathcal{A}_g$ makes a query to $\mathcal{H}$ containing m_d, $\mathcal{B}'_g$ outputs d as its guess. Note that the issue of correctly simulating the random oracle does not arise here, as $\mathcal{B}'_g$ aborts as soon as $\mathcal{A}_g$ makes a critical query. Analogously, we show that when $b = 0$, the probability that $\mathcal{A}_g$ queries $(m_1 \| \cdot)$ is negligible.

It remains to prove that when $b = 1$ then $\mathcal{A}_g$ queries $(m_1 \| \cdot)$ with negligible probability. Again, assuming $\mathcal{A}_g$ makes such a query, we construct $\mathcal{B}''$ breaking ind-cpa. As before, $\mathcal{B}''$ uses $\mathcal{A}_f$ to sample m_0 and m_1 and receives c. Now $\mathcal{B}''_g$ picks a random bit d and sends $\mathcal{A}_g$ the following: (c, r_0), for some random r_0 and a PCE encryption of m_d. If $\mathcal{A}_g$ queries $(m_d \| \cdot)$ then $\mathcal{B}''_g$ outputs d. (Note that up to this point, the simulation is perfect.) We show that $\mathcal{B}''$ wins the indistinguishability game. If d equals $\mathcal{B}''$'s challenger's bit then $\mathcal{A}_g$ gets two encryptions of the same message; $\mathcal{A}$ is thus playing the unlink game with $b = 1$, for which we assumed $\mathcal{A}_g$ queries the encrypted message to $\mathcal{H}$ with non-negligible probability, in which case $\mathcal{B}''$ wins. On the other hand, if d is different from the challenger's bit (in which case $\mathcal{B}''$ loses) then $\mathcal{A}$ gets encryptions of two different messages and it is thus playing unlink with $b = 0$. For this case however, the previous result for $b = 1$ asserts that $\mathcal{A}$ will not query an encrypted message to the random oracle.

Appeared in Orr Dunkelman (Ed.): CT-RSA 2012, volume 7178 of LNCS, pp. 332-348, 2012.

Algorithm $\mathsf{KeyGen}(1^k)$

$(\overline{pk}, \overline{sk}) \xleftarrow{\$} \Pi_d.\mathcal{G}_d(1^k)$
$pk \leftarrow \overline{pk}$
$sk \leftarrow \overline{sk}$
return (pk, sk)

Algorithm $\mathsf{Encrypt}(1^k, pk, m)$

$\overline{pk} \leftarrow pk$
$r \xleftarrow{\$} \{0,1\}^{\ell(k)}$
$\rho \leftarrow \mathcal{H}_1(m\|r)$
$\overline{c_1} \leftarrow \Pi_d.\mathcal{E}_d(1^k, \overline{pk}, \rho)$
$\overline{c_2} \leftarrow m \oplus \mathcal{H}_2(\rho)$
$C \leftarrow (\overline{c_1}, \overline{c_2}, r)$
return C

Algorithm $\mathsf{Decrypt}(1^k, sk, C)$

$(\overline{c_1}, \overline{c_2}, r) \leftarrow C$
$\overline{sk} \leftarrow sk$
$\rho \leftarrow \Pi_d.\mathcal{D}_d(1^k, \overline{sk}, \overline{c_1})$
$m \leftarrow \overline{c_2} \oplus \mathcal{H}_2(\rho)$
if $\rho = \mathcal{H}_1(m||r)$ then return m

Algorithm $\mathsf{PCheck}(1^k, pk, C, m)$

$(\overline{c_1}, \overline{c_2}, r) \leftarrow C$
$\overline{pk} \leftarrow pk$
$\rho \leftarrow \mathcal{H}_1(m\|r)$
$\tilde{c} \leftarrow \Pi_d.\mathcal{E}_d(1^k, \overline{pk}, \rho)$
if $\tilde{c} = \overline{c_1}$ then return 1
else return 0

Fig. 3. Unlinkable PCE from a deterministic encryption scheme Π_d

3.2 A PCE based on a Deterministic Encryption Scheme

Let $\Pi_d = (\mathcal{G}_d, \mathcal{E}_d, \mathcal{D}_d)$ be a secure deterministic encryption scheme, meaning that it satisfies the ind-det property as defined in [4] and recalled in Sect. 2.2. Let $\mathcal{H}_1 : \{0,1\}^* \to \{0,1\}^{\ell(k)}$ and $\mathcal{H}_2 : \{0,1\}^* \to \{0,1\}^{\ell(k)}$ be two hash functions modeled as random oracles.

The idea behind this construction is to encrypt with the deterministic encryption algorithm a hash value ρ of the message m together with a random element r and then to compute a one-time pad of the message and the hash value of ρ. We include r in the ciphertext, so knowing m and r, one can recompute the (deterministic) ciphertext and thus perform the plaintext check.

Our random-oracle based construction is detailed in Fig. 3, and Corollary 1 states its security. As we will see, this theorem is a consequence of Theorem 1.

Corollary 1 (sketch, see full version for the full proof). *The PCE construction given in Fig. 3 is unlinkable under the assumption that Π_d is one-way, in the random-oracle model.*

Proof (sketch, see full version for the full proof). This proof is a direct application of Theorem 1 combined with the result from [5] which states that the encryption scheme which consists in computing $c_1 \leftarrow \Pi_d.\mathcal{E}(1^k, pk, r)$ and $c_2 \leftarrow m \oplus \mathcal{H}_2(r)$, where $r \xleftarrow{\$} \{0,1\}^{\ell(k)}$, is ind-cpa if the underlying deterministic encryption scheme Π_d is one-way. □

4 Practical Constructions in the Standard Model

A construction of a secure plaintext-checkable encryption can be proved in the standard model using the technique from [4] for deterministic encryption (see

Appeared in Orr Dunkelman (Ed.): CT-RSA 2012, volume 7178 of LNCS, pp. 332-348, 2012.

Fig. 3 of [4]): one replaces the random oracle by a pseudo-random generator [7, 24, 15] based on a family of trapdoor permutations. As for the previous construction, the idea is to use a secure encryption scheme whose randomness is generated using a secure pseudo-random generator with a seed depending on the message and the random value used to check the plaintext. We here give another practical construction based on the ElGamal encryption scheme [14], which we will then use for our standard-model VLR group signature scheme given in Sect. 5.

4.1 An ElGamal-Based Construction

Our construction lies in an asymmetric bilinear group $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, h)$ where p is a large prime, $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are cyclic groups of order p and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a non-degenerate bilinear map. The elements g and h denote generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. In our scheme, the idea is to encrypt a message m under a public key y using randomness r as $c_1 = my^r, c_2 = g^r$. If we gave $c_3 = h^r$ as well, then using the pairing we can perform plaintext checks since $e(c_1 m^{-1}, g) = e(y, c_3)$. However, this construction does not achieve unlinkability, since we can check whether 2 ciphertexts encrypt the same message by checking whether their quotient encrypts 1. To avoid this, instead of using h as a base for the check element c_3, we use a random base h^a. Since this base is different for every ciphertext, no two ciphertexts can be combined. Our construction is described in Fig. 4 and allows to encrypt messages $m \in \mathbb{G}_1$.

Algorithm $\mathsf{KeyGen}(1^k)$

$x \xleftarrow{\$} \mathbb{Z}_p^*$
$y \leftarrow g^x$
$(pk, sk) \leftarrow (y, x)$
return (pk, sk)

Algorithm $\mathsf{Encrypt}(1^k, pk, m)$

$y \leftarrow pk$
$r, a \xleftarrow{\$} \mathbb{Z}_p^*$
$C \leftarrow (my^r, g^r, h^a, h^{ar})$
return C

Algorithm $\mathsf{Decrypt}(1^k, sk, C)$

$x \leftarrow sk$
$(c_1, c_2, c_3, c_4) \leftarrow C$
if $e(g, c_4) \neq e(c_2, c_3)$ then return $\perp$
$m \leftarrow c_1 / c_2^x$
return m

Algorithm $\mathsf{PCheck}(1^k, pk, C, m)$

$y \leftarrow pk$
$(c_1, c_2, c_3, c_4) \leftarrow C$
if $e(g, c_4) \neq e(c_2, c_3)$ then return 0
if $e(c_1/m, c_3) = e(y, c_4)$ then return 1
else return 0

Fig. 4. Unlinkable PCE in the standard model

4.2 Security Arguments

To prove unlinkability of the construction in Fig. 4, we introduce a new assumption (whose security in the generic-group model is proved in the full version), which combines features of the Decision Linear Assumption (DLIN) and the assumption that DDH holds in both base groups of an asymmetric bilinear group (known as "SXDH").

Appeared in Orr Dunkelman (Ed.): CT-RSA 2012, volume 7178 of LNCS, pp. 332-348, 2012.

Assumption 1 *Given an asymmetric bilinear group* $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ *with generators* $g \in \mathbb{G}_1$ *and* $h \in \mathbb{G}_2$*, and the tuple* $(g^x, g^{rx}, g^{sx}, h^a, h^{ar}, h^b, h^{br}, V)$ *for random* $x, r, s, a, b \in \mathbb{Z}_p$*, it is hard to decide if* $V = g^{r+s}$ *or* V *is random in* $\mathbb{G}_1$*.*

Let us first analyze the $\mathbb{G}_1$ part of our assumption: (g^x, g^{rx}, g^{sx}) and g^{r+s}. DLIN states that given $(g^x, g^y, g^{rx}, g^{sy})$ it is hard to distinguish g^{r+s} from random. The $\mathbb{G}_1$ components of our assumption can thus be seen as a DLIN instance with $y = x$ (note that whereas DLIN also holds in *symmetric* groups, this is not the case when $y = x$). It is also immediate that this "partial" assumption is a DDH instance where $s = 0$, and thus implied by DDH. However, since—as opposed to DDH—we have *two* random *combined* exponents r and s for the challenge, this allows us to add values depending on them in $\mathbb{G}_2$, which cannot be used to verify the structure of g^{r+s}, since the bases h^a and h^b for r and s are different.

The following theorem holds against adversaries $\mathcal{A} = (\mathcal{A}_f, \mathcal{A}_g)$ where $\mathcal{A}_f$ outputs the uniform distribution. This restriction is similar to the results by Bellare et al. [3] for their practical construction of a deterministic encryption scheme. In fact, in real life applications, the uniform distribution is most of time enough and easily obtained. In particular, this notion also suffices when applying the scheme to VLR group signatures.

Theorem 2. *Under Assumption 1, the construction from Fig. 4 is a PCE scheme which is* unlink *against adversaries outputting the uniform distribution.*

5 Application to VLR Group Signature

In this section we use our new primitive as a building block for group signatures with verifier-local revocation (VLR) [10]. This is a group signature scheme [2, 8] which allows an efficient revocation of group members.

Our aim in this section is twofold. First, we present plaintext-checkable encryption as a new building block for group signatures with VLR; thus any improvement to PCE is likely to lead to more efficient group signatures with VLR. Second, we design in the following, to the best of our knowledge, the most efficient group signature scheme with VLR and backward unlinkability in the standard model. We first recall the concept of group signatures with VLR, and eventually describe our new construction.

5.1 Definitions for Group Signatures with VLR

Let k, n and T be integers. A *group signature scheme with VLR* (VLR-GS for short) is composed of the following algorithms (following [19]).

- KeyGen takes as input a security parameter 1^k, the number n of group members and the number T of time periods. It produces the group public key gpk, an n-element vector of user keys $\mathbf{sk} = (\mathsf{sk}_1, \ldots, \mathsf{sk}_n)$ and an $(n \times T)$-element vector of user revocation tokens $\mathbf{grt} = (\mathsf{grt}[1][1], \ldots, \mathsf{grt}[n][T])$.

Appeared in Orr Dunkelman (Ed.): CT-RSA 2012, volume 7178 of LNCS, pp. 332-348, 2012.

– Sign takes as input the group public key gpk, the current time interval j, a secret key sk_i for $i \in [\![1, n]\!]$ of a group member and a message $m \in \{0,1\}^*$, and outputs a signature σ.
– Verify takes as input the group public key gpk, the current time period j, the public key of the revocation authority rpk, a set of revocation tokens RL_j, and a purported signature σ on a message m. It returns either valid if the signature σ is valid or invalid if σ is not a valid signature or if the user who generated it has been revoked.

The security requirements are *traceability* and *backward unlinkability (BU) anonymity*. The corresponding formal definitions can be found in [19]. We only recall the BU-anonymity since adding the VLR functionality to a group signature scheme only concerns this security notion, whereas traceability is inherited from the original scheme. A VLR-GS with backward unlinkability is BU-anonymous if no p.p.t. adversary $\mathcal{A}$ has non-negligible advantage in the following game.

1. The challenger $\mathcal{C}$ executes $(\mathsf{gpk}, \mathbf{sk}, \mathbf{grt}) \xleftarrow{\$} \mathsf{KeyGen}(1^k, n, T)$ and the adversary is given gpk.
2. For each period, $\mathcal{C}$ increments the counter j and during this period, $\mathcal{A}$ can access the $\mathsf{Sign}(\cdot,\cdot)$ oracle, which gives a group signature on a message m by a user i during time period j, the $\mathsf{Corrupt}(\cdot)$ oracle, which permits to corrupt the user i and the $\mathsf{Revoke}(\cdot)$ oracle, which revokes the member i.
3. At some period $j^* \in [1, T]$, $\mathcal{A}$ outputs (m^*, i_0, i_1) such that i_0 and i_1 are not corrupted and have not been revoked during or before the time period j^*. The challenger $\mathcal{C}$ flips a coin b and generates $\sigma^* \xleftarrow{\$} \mathsf{Sign}(\mathsf{gpk}, j^*, \mathsf{sk}_{i_b}, m^*)$, which is sent to $\mathcal{A}$.
4. $\mathcal{A}$ can again access the above oracles. $\mathcal{A}$ is not allowed to corrupt i_0 nor i_1 but it may revoke them after time period j^*.
5. Eventually, $\mathcal{A}$ outputs a bit b^* and wins if $b = b^*$.

The advantage of $\mathcal{A}$ in breaking this anonymity is defined as $\mathbf{Adv}^{\mathsf{bu\text{-}a}}_{\mathsf{VLR\text{-}GS},\mathcal{A}}(k) := |\Pr[b = b^*] - \frac{1}{2}|$.

5.2 Using PCE for Group Signatures with VLR

Starting with a group signature scheme. For concreteness, we base our instantiation on the group signature scheme by Fuchsbauer and Abe et al. in [12, 1], which is itself based on Groth's scheme [16], which makes use of the non-interactive zero-knowledge (NIZK) proofs from Groth and Sahai [17].

In a nutshell, each user creates a key pair for an *automorphic signature* scheme[3] [12, 1]. The group public key is a signature verification key, whose corresponding signing key is used by the group manager to sign a user's verification

[3] A signature scheme defined over a bilinear group is *automorphic* if the verification keys lie in the message space, and if the messages and the signatures consist of group elements. The first property enables certification of keys, whereas the second makes it possible to give efficient NIZK proofs of knowledge of valid signatures and messages using Groth-Sahai proofs.

Appeared in Orr Dunkelman (Ed.): CT-RSA 2012, volume 7178 of LNCS, pp. 332-348, 2012.

key when he joins the group. To make a group signature, the user first signs the message using his personal signing key; the group signature is then a Groth-Sahai proof of knowledge of the following: the user's verification key, a valid certificate on it by the group manager, and a signature on the message that is valid under his verification key. Since the registration protocol consists of only one round, the scheme is concurrently secure. Moreover, since the group members create their own signing keys, the scheme achieves *non-frameability* [6].

Adding the VLR property. When adding verifier-local revocability, to achieve backward unlinkability, we use the system due to Nakanishi and Funabiki [20]. This consists in defining time periods and constructing one key (called the revocation token) per group member and time period. This token is to be used by the group member when making a group signature. When a member is revoked, all the revocation tokens related to the revoked group member and future time periods are published. These public revocation tokens are then used by the verifier to check whether the received group signature has been produced with a published value, and thus by a revoked group member.

Making use of a PCE. The group signature cannot contain the revocation token in the clear, as this would compromise the member's anonymity. Our approach is to include in the group signature a *plaintext-checkable encryption* of the revocation token, together with a proof of well-formedness. When a revoked group member's token gets published, the verifier can use PCheck of the PCE scheme to check whether the group signature comes from a revoked member or not. For our concrete scheme, we use the standard-model PCE scheme from Sect. 4.1, since it complies with the Groth-Sahai methodology.

5.3 Our Concrete Instantiation

We will use the group signature scheme on which we base our construction as a black box and simply add one PCE encryption and a proof of consistency to make it a VLR scheme. We require that the group signature is a Groth-Sahai proof of knowledge in an asymmetric bilinear group $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, h)$ and that the user verification key contains a component h^{v_i}, where v_i is the i-th user's signing key. (This is the case e.g. in the construction from [12, 1]).

In the setup phase of scheme (when the common reference string for Groth-Sahai proofs is created), we now also create a key pair $(y = g^x, x) \in \mathbb{G}_1 \times \mathbb{Z}_p$ for our PCE scheme from Sect. 4.1 and add y to the public parameters. As in [20, 19], we introduce a vector $(P_1, \ldots, P_T)$ of $\mathbb{G}_1$ elements, where T is the maximum number of time periods. The revocation token for user i (holding secret key v_i) for time interval j is defined as $P_j^{v_i}$.

When creating a group signature, the user must additionally encrypt his token for the current time interval and prove that it is well-formed. The token is of the form P^v, so we need to prove that v is the same as in the user verification key element $w := h^v$ (of which the group signature will prove knowledge). The PCE encryption of the token is $C = (C_1, C_2, C_3, C_4) = (P^v y^r, g^r, h^a, h^{ar})$. To prove well-formedness, we introduce an auxiliary variable $z := h^r$, of which

Appeared in Orr Dunkelman (Ed.): CT-RSA 2012, volume 7178 of LNCS, pp. 332-348, 2012.

we also prove knowledge in the group signature. Groth-Sahai proofs allow us to prove knowledge of group elements that satisfy *pairing-product equations* (PPE). Let v be such that P^v is the plaintext of C. Then the following PPEs assert that $w = h^v$ (the group elements of which we prove knowledge are underlined): $e(C_1, h) = e(P, \underline{w})\, e(y, \underline{z})$ and $e(C_2, h) = e(g, \underline{z})$.

In addition to C, we include in the group signature a Groth-Sahai NIZK proof that the above equations are satisfied. Our new verification procedure now additionally checks this new proof component, and runs PCheck on C and the elements of the revocation list to check if the user has been revoked.

We note that our techniques also work if the verification key contains g^v rather than h^v: we can introduce a second encrypted auxiliary variable $z' := h^v$ and add a proof of $e(\underline{g^v}, h) = e(g, \underline{z'})$. We have thus shown that adding to a Groth-Sahai based group signature scheme (with user verification keys containing a generator to the power of the signing key) a plaintext-checkable encryption of a token, gives a group signature scheme with VLR and backward unlinkability.

5.4 Backward-Unlinkable Anonymity

We outline the proof that our scheme satisfies backward-unlinkable anonymity. The proof proceeds by a series of games. The first game is the experiment defined in Sect. 5.1. In the second game, instead of running KeyGen, we compute the common reference string for Groth-Sahai proofs in a way that will lead to perfectly hiding proofs of knowledge, which can be simulated. By the zero-knowledge property of Groth-Sahai proofs, the first two games are indistinguishable. In Game 3, the challenger picks 2 random users, hoping they will be the challenge users i_0 and i_1 output by the adversary in Step 3 of the game. If the challenger did not guess these users correctly, it aborts the game. This introduces a polynomial loss in the security reduction.

In Game 4 the challenger simulates the NIZK proofs in the following signatures it gives to the adversary: all signatures in signing queries for users i_0 and i_1 queried up to the challenge time period j^*; and the challenge signature σ^*. It follows from the zero-knowledge property of Groth-Sahai proofs that Game 4 is indistinguishable from Game 3.

We can now play with the plaintext-checkable encryptions C of tokens which are given to the adversary as part of the simulated group signature (either in a signing query for users i_0 and i_1 in time $j < j^*$ or the challenge signature). Since the proof of consistency of these C's is simulated, we can change the actual values, which we will do in the following. Next, when computing the values P_j during setup, the challenger sets them as $P_j := g^{d_j}$ and stores d_j. We now define a series of games, in which, one by one, we replace tokens $P_1^{v_{i_0}}, \ldots, P_{j^*}^{v_{i_0}}$ and tokens $P_1^{v_{i_1}}, \ldots, P_{j^*}^{v_{i_1}}$ by random values. This is reduced to the DDH assumption, which implies that given values g^d and g^v, we can replace g^{dv} by a random value. Note that given a DDH challenge, the challenger can use the logarithms d_j to compute the values $P_j^{v_i}$ it is not changing in that step.

After this series of games, the only dependency of the challenge signature on the bit b occurs when the adversary asks for a signature of user i_b in time

Appeared in Orr Dunkelman (Ed.): CT-RSA 2012, volume 7178 of LNCS, pp. 332-348, 2012.

interval j^*. Since the tokens are chosen uniformly at random, we can replace the encryption of the token in the challenge signature by a random value. This is implied by unlinkability of our PCE scheme (which states that two encryptions of the same value are indistinguishable from two encryptions of two different (random) values). After this final step the challenge signature is independent of b and the adversary's winning probability is thus exactly $\frac{1}{2}$.

5.5 Comparison with Related Work

Regarding related work on group signature schemes with VLR, there are typically 3 criteria to compare such schemes: random-oracle or standard model, anonymity revocation or not and backward unlinkability or not. Table 1 compares all existing solutions, to the best of our knowledge.

Table 1. Related work on group signatures with VLR

Papers	Standard model	Anonymity revocation	Backward unlinkability
[10]	No	No	No
[20, 21, 25]	No	Yes	Yes
[19]	Yes	(Yes)	Yes
Ours	Yes	(Yes)[4]	Yes

Achieving CCA security. An additional property not considered in the above table is CCA-anonymity, meaning the scheme remains anonymous even if the adversary has an oracle to open signatures of its choice, as considered e.g. in the model by Bellare et al. [6]. This notion is achieved by variants of the group signature schemes on which we base our VLR scheme, using one-time signatures and a weakly CCA tag-based encryption scheme, as proposed by Groth in [16].

The tag-based encryption scheme used is Kiltz's construction [18] is secure under the DLIN assumption [8] and is defined over symmetric bilinear groups. As DDH is easy in such groups, our PCE scheme would not be secure and can thus not be added to these schemes. We believe however that starting from *linear encryption* [8] rather than ElGamal, and adding elements enabling plaintext checkability, one could define a PCE scheme over *symmetric* bilinear groups.

Efficiency considerations. We can now compare the efficiency of standard model group signatures with VLR and backward unlinkability, which amounts to comparing us with the scheme by Libert and Vergnaud [19]. On one hand, regarding [19], a group signature is composed of 46 elements in $\mathbb{G}$ and 1 element in $\mathbb{G}_T$. The time complexity of a group-signature creation necessitates 2 modular exponentiations in $\mathbb{G}$, 6 commitment generations, 2 quadratic GS proofs

[4] Not explicitly detailed but can be easily added by giving the trapdoor for the CRS of Groth-Sahai proofs to the opener.

Appeared in Orr Dunkelman (Ed.): CT-RSA 2012, volume 7178 of LNCS, pp. 332-348, 2012.

and 4 linear GS proofs. The revocation checking requires the computation of one pairing per element in RL_j. On the other hand, our signatures are composed of 12 elements in $\mathbb{G}_1$, 18 elements in $\mathbb{G}_2$ and no element in $\mathbb{G}_T$. The signer must perform 6 modular exponentiations, 1 quadratic GS proofs and 5 linear GS proofs. The revocation checking requires the computation of 2 pairings per element in RL_j. Considering moreover that in asymmetric groups, representations of group elements are shorter and computation of pairings are much more efficient, our scheme is more efficient in terms of signature computation and size but necessitates slightly more work during the revocation check.

6 Conclusion

We proposed a new promising public-key encryption scheme with a special feature: this primitive allows anyone to verify whether a given ciphertext (together with the public key used to encrypt) actually encrypts any potential message. However, if the messages come from a space with enough entropy, one cannot decide whether two ciphertexts encrypt the same message. Plaintext-checkable encryption with unlinkable ciphertexts is perfectly adapted to design group signatures with verifier-local revocation and backward unlinkability. The efficiency of the constructions also enables its use in a context of cloud storage services.

Acknowledgements: This work has been supported by the French Agence Nationale de la Recherche under the PACE 07 TCOM Project, the European Commission under Contract ICT-2007-216676 ECRYPT II and EPSRC Grant EP/H043454/1. We are grateful to Jacques Traoré for his suggestions of improvement, and to the anonymous referees for their valuable comments.

References

1. Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Structure-preserving signatures and commitments to group elements. In *Advances in Cryptology - CRYPTO 2010*, volume 6223 of *LNCS*. Springer, 2010.
2. Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *Proc. of Crypto 2000*, volume 1880 of *LNCS*, pages 255–270. Springer, 2000.
3. Mihir Bellare, Alexandra Boldyreva, and Adam O'Neill. Deterministic and efficiently searchable encryption. In *Proc. of Crypto 2007*, volume 4622 of *LNCS*, pages 535–552. Springer, 2007.
4. Mihir Bellare, Marc Fischlin, Adam O'Neill, and Thomas Ristenpart. Deterministic encryption: Definitional equivalences and constructions without random oracles. In *Proc. of Crypto 2008*, volume 5157 of *LNCS*, pages 360–378. Springer, 2008.
5. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73, 1993.
6. Mihir Bellare, Haixia Shi, and Chong Zhang. Foundations of group signatures: The case of dynamic groups. In *Proc. of CT-RSA 2005*, volume 3376 of *LNCS*, pages 136–153. Springer, 2005.

Appeared in Orr Dunkelman (Ed.): CT-RSA 2012, volume 7178 of LNCS, pp. 332-348, 2012.

7. Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM J. Comput.*, 13(4):850–864, 1984.
8. Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *Proc. of Crypto 2004*, volume 3027 of *LNCS*, pages 41–55. Springer, 2004.
9. Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 506–522. Springer, 2004.
10. Dan Boneh and Hovav Shacham. Group signatures with verifier-local revocation. In *ACM Conference on Computer and Communications Security*, pages 168–177. ACM, 2004.
11. Jan Camenisch and Jens Groth. Group signatures: Better efficiency and new theoretical aspects. In *SCN 2004*, volume 3352 of *LNCS*, pages 120–133. Springer, 2004.
12. Georg Fuchsbauer. Automorphic signatures in bilinear groups and an application to round-optimal blind signatures. Cryptology ePrint Archive, Report 2009/320, 2009. http://eprint.iacr.org/.
13. Thomas Fuhr and Pascal Paillier. Decryptable searchable encryption. In *Proc. of ProvSec 2007*, volume 4784 of *LNCS*, pages 228–236. Springer, 2007.
14. Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.
15. Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *Proc. of STOC'89*, pages 25–32. ACM, 1989.
16. Jens Groth. Fully anonymous group signatures without random oracles. In *Proc. of Asiacrypt 2007*, volume 4833 of *LNCS*, pages 164–180. Springer, 2007.
17. Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In *Proc. of Eurocrypt 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, 2008.
18. Eike Kiltz. Chosen-ciphertext security from tag-based encryption. In *TCC 2006*, volume 3876 of *LNCS*, pages 581–600. Springer, 2006.
19. Benoît Libert and Damien Vergnaud. Group signatures with verifier-local revocation and backward unlinkability in the standard model. In *Proc. of CANS 2009*, volume 5888 of *LNCS*, pages 498–517. Springer, 2009.
20. Toru Nakanishi and Nobuo Funabiki. Verifier-local revocation group signature schemes with backward unlinkability from bilinear maps. In *ASIACRYPT'05*, volume 3788 of *LNCS*, pages 533–548. Springer, 2005.
21. Toru Nakanishi and Nobuo Funabiki. A short verifier-local revocation group signature scheme with backward unlinkability. In *IWSEC*, volume 4266 of *LNCS*, pages 17–32. Springer, 2006.
22. Rafail Ostrovsky and William E. Skeith III. Private searching on streaming data. *J. Cryptology*, 20(4):397–430, 2007.
23. Guomin Yang, Chik How Tan, Qiong Huang, and Duncan S. Wong. Probabilistic public key encryption with equality test. In *Proc. of CT-RSA 2010*, volume 5985 of *LNCS*, pages 119–131. Springer, 2010.
24. Andrew Chi-Chih Yao. Theory and applications of trapdoor functions (extended abstract). In *Proc. of FOCS'82*, pages 80–91. IEEE, 1982.
25. Sujing Zhou and Dongdai Lin. Shorter verifier-local revocation group signatures from bilinear maps. In *CANS 2006*, volume 4301 of *LNCS*, pages 126–143. Springer, 2006.

Appeared in Orr Dunkelman (Ed.): CT-RSA 2012, volume 7178 of LNCS, pp. 332-348, 2012.