Verifiable Predicate Encryption and Applications to CCA Security and Anonymous Predicate Authentication

Shota Yamada*[1],Nuttapong Attrapadung[2], Bagus Santoso[3],
Jacob C.N. Schuldt**[2], Goichiro Hanaoka[2], and Noboru Kunihiro[1]

[1] The University of Tokyo. {yamada@it., kunihiro@} k.u-tokyo.ac.jp
[2] National Institute of Advanced Industrial Science and Technology (AIST).
{n.attrapadung, jacob.schuldt, hanaoka-goichiro}@aist.go.jp
[3] Institute for Infocomm Research. santosob@i2r.a-star.edu.sg

Abstract. In this paper, we focus on *verifiability* of predicate encryption. A verifiable predicate encryption scheme guarantees that all legitimate receivers of a ciphertext will obtain the same message upon decryption. While verifiability of predicate encryption might be a desirable property by itself, we furthermore show that this property enables interesting applications.

Specifically, we provide two applications of verifiable predicate encryption. Firstly, we show that for a large class of verifiable predicate encryption schemes, it is always possible to convert a chosen-plaintext secure scheme into a chosen-ciphertext secure one. Secondly, we show that a verifiable predicate encryption scheme allows the construction of a *deniable predicate authentication scheme*. This primitive enables a user to authenticate a message to a verifier using a private key satisfying a specified relation while at the same time allowing the user to deny ever having interacted with the verifier. This scheme furthermore guarantees the anonymity of the user in the sense that the verifier will learn nothing about the user's private key except that it satisfies the specified relation. Lastly, we show that many currently known predicate encryption schemes already provide verifiability, and furthermore demonstrate that many predicate encryption schemes which do not provide verifiability, can be easily converted into schemes providing verifiability.

Our results not only highlight that verifiability is a very useful property of predicate encryption, but also show that efficient and practical schemes with this property can be obtained relatively easily.

1 Introduction

In many practical data transmission systems, we often encounter situations in which a sender would like to securely transmit data to a set of users satisfying certain criteria. To address this, several frameworks and concrete instantiations

* The first author is supported by JSPS Research Fellowships for Young Scientists.
** The fourth author is supported by JSPS Research Fellowships for Young Scientists.

providing *encryption with multiple receivers* have been proposed in the literature. Examples of such frameworks include broadcast encryption (BE) [8, 35], spatial encryption [9, 2], and ciphertext-policy/key-policy attribute-based encryption (CP/KP-ABE) [17, 7, 36, 22, 21, 28]. All of the above mentioned schemes can be seen as special cases of *predicate encryption* (PE) which is a new emerging paradigm for public key encryption that allows a fine-grained access control mechanism to be specified for encrypted data. More specifically, in an PE scheme for relation R, a ciphertext will be associated with a ciphertext attribute Y while a private key corresponds to key attribute X, and the decryption can be done only if the relation $R(X, Y)$ is satisfied. In this paper, we consider a wide class of relations which covers the above mentioned special cases.

Previous works on PE have mainly focused on security properties regarding privacy, namely *message privacy* (also referred to as payload hiding) and *ciphertext attribute hiding* (also referred to as anonymity). The former captures the property that a ciphertext with attribute X reveals no information about the encrypted messages if one does not possess a key with attribute Y such that $R(X, Y)$ is satisfied. The latter captures the property that for anyone in the possession of a private key with attribute X', a ciphertext reveals no information about the ciphertext attribute Y other than what is implied by $R(X', Y)$.

In this paper, we focus on *verifiability* of PE and the applications of PE schemes providing this property. If an PE scheme provides verifiability, it is guaranteed that all legitimate receivers of a ciphertext will obtain the same message upon decryption i.e. for a ciphertext with attribute Y, the decryption using two different private keys corresponding to attributes X and X' where both $R(X, Y)$ and $R(X', Y)$ are satisfied, will always yield the same message. Verifiability in itself is arguably a useful property and might even be required for some applications. For example, in pay-per-view systems, receivers might demand to be able to confirm that decryption results among all paying receivers are identical, especially in the case the decryption result is different from the expected. This property is guaranteed if an PE scheme with verifiability is used to broadcast data to the receivers.

However, besides guaranteeing consistency of the decryption results among all legitimate receivers, verifiability will furthermore enable interesting applications of PE schemes providing this property. In this paper, we show two specific applications of verifiable predicate encryption (VPE). More specifically, firstly we show that it is always possible to convert an arbitrary chosen-plaintext secure (CPA-secure) VPE with an arbitrary flavor into chosen-ciphertext secure (CCA-secure) one with the same flavor. For example, if it is possible to show that a CPA-secure spatial encryption scheme provides verifiability (this is, for example, the case for the spatial encryption scheme presented in [9]), we immediately obtain a CCA-secure spatial encryption scheme. One might think that this can easily be achieved by applying the Canetti-Halevi-Katz [12, 10] technique. However, it is unclear whether this technique can be adapted to PE in general, and specifically, for concrete special cases of PE such as inner product encryption and broadcast encryption, the Canetti-Halevi-Katz technique cannot be applied in a straight forward manner. The method applied in our conversion is closer related to the Naor-Yung technique [26]. We also remark that the techniques presented in this paper can be seen as a non-trivial generalization of the technique presented in [37].

Secondly, we show that a VPE scheme allows the construction of an anonymous deniable predicate authentication (ADPA) scheme. This primitive enables a user to prove to a verifier that he is the owner of a private key corresponding to a specific set of attributes while at the same time being able to deny ever having interacted with the verifier. More specifically, for a ciphertext attribute Y, possibly chosen at the time of authentication, a user can authenticate a message to a verifier using a private key with attribute X such that $R(X, Y)$ is satisfied. The deniability property furthermore guarantees that the verifier's view of the communication can be produced a posteriori without the knowledge of the private key corresponding to X. Hence, the transcript of the interaction cannot be used as evidence of the user authenticating the message to the verifier, and the user will be able to deny ever having done so. ADPA will furthermore guarantee anonymity of the user in the sense that the verifier will not be able to determine the attribute X of the private key of the user, but will only be able to verify that $R(X, Y)$ is satisfied. In other words, the verifier will be able to confirm that the user belongs to the set of users to which the key authority issued a key with property X such that $R(X, Y)$ is satisfied. This anonymity property is guaranteed to hold even if verifier collude with the authority issuing the private keys of the users. While not being directly comparable, this is reminiscent of the properties provided by anonymous credentials which allows a user to demonstrate knowledge of a credential issued by an authority, but without revealing his identity.

Lastly, we show that many concrete PE schemes already provide verifiability, and furthermore demonstrate that many PE schemes which do not provide verifiability, can be easily converted into schemes providing verifiability. Our conversion techniques are applicable to a wide range of (non-verifiable) PE schemes. As examples, we briefly discuss how Waters BE scheme [35], Attrapadung-Libert inner product encryption (IPE) scheme and spatial encryption scheme [1], and Okamoto-Takashima KP-ABE schemes [28] can be transformed into schemes providing verifiability by introducing only simple modifications. This shows that efficient and practical VPE scheme can be constructed, which, due to the results presented in this paper, implies that efficient and CCA-secure variants of these schemes can be obtained as well.

Related Works on PE. In its simplest form, PE corresponds to id-based encryption [5, 4, 34]. Sahai and Waters [33] proposed the first ABE system with much more expressive relations called *Fuzzy IBE*. It was subsequently generalized to support general access policies by [17, 7, 29, 22]. These results are proved secure in a weak model called selective security. The first fully-secure ABE systems were given by Lewko et al. [21] and Okamoto and Takashima [28], following the general dual-system encryption methodology introduced in [35, 24]. When efficiency is the main consideration, the first system with constant-size ciphertexts and with reasonably expressive policies was proposed by Boneh and Hamburg [9], where a system called *spatial encryption* was presented. A fully-secure scheme for spatial relations was then proposed by Attrapadung and Libert [2], where its extension to support inner-product, of which many applications such as CNF/DNF formulae policy expressions as described in [20], was also given. All the aforementioned systems so far do not concern the security property regarding the privacy of ciphertext attributes. The first *attribute-hiding* predicate encryption, or equivalently known as *functional encryption* was suggested by Boneh and Waters [11] and generalized by Katz, Sahai, and Waters [20] to support inner product relations. These attribute-hiding systems were recently made fully secure in [21, 28].

Related Works on Deniable Authentication. The formal treatment of *deniability* for public key authentication was initiated by Dwork, Naor and Sahai in their paper on concurrent zero-knowledge[13], followed by a series of papers [25, 30, 31]. In [13], Dwork et al. propose a deniable authentication protocol based on a CCA-secure encryption scheme. Naor [25] later extended the work by Dwork et al., and introduced the concept of *deniable ring authentication* by combining the approach of Dwork et al. and the paradigm of *ring signatures* proposed by Rivest et al. [32]. In a deniable ring authentication, a member of a ring can authenticate a message in a deniable way to a receiver. Another approach not relying on CCA-secure encryption scheme was proposed by Raimondo and Gennaro [31]. They successfully eliminate the need for CCA-secure encryption by using another primitive, i.e., multi-trapdoor commitments [14]. It should be noted that all these works are in the plain model. Meanwhile, Pass in [30] investigates the possibility of constructing deniable zero-knowledge protocols in the non-plain models, i.e., the common reference string model and random oracle model. Pass shows an impossibility result regarding the construction of non-trivial deniable zero-knowledge protocols in the common reference string model, and a positive result, in the random oracle model, regarding the construction of efficient deniable zero-knowledge arguments of knowledge which preserve both the zero-knowledge property and the proof of knowledge property under concurrent executions.

Notations. $a \overset{\$}{\leftarrow} A$ denotes the action of picking a from uniform random distribution over A. $negl(\lambda)$ denotes negligible function in λ. $A \overset{c}{\approx} B$ denotes A and B are computationally indistinguishable. $[A(x)]$ for randomized algorithm A and its input x denotes a set $\{y | \Pr[A(x) = y] \neq 0\}$.

2 Definition of Verifiable Predicate Encryption

In this section we introduce the definition and security notion for PE, and furthermore introduce verifiability. Note that our definition of verifiability is similar but slightly different from the definitions given in [37]. More specifically, the definition given in [37] explicitly requires a **Verify** algorithm, whereas our definition defines verifiability as a property of the decryption algorithm. Thus, our definition of the verifiability is more similar to that of [18]. We also note that our definition of verifiability is orthogonal to the notion defined in [6].

2.1 Definition of Predicate Encryption

Here, we define the notion of predicate encryption.
Syntax. Let $R = \{R_n : A_n \times B_n \to \{0, 1\} \mid n \in \mathbb{N}\}$ be a relation family where A_n and B_n denote "key attribute" and "ciphertext attribute" spaces. A predicate encryption (PE) scheme for R consists of the following algorithms:

Setup$(\lambda, n) \to (PK, MSK)$: The setup algorithm takes as input a security parameter λ and a dimension n of the relation R and outputs a public key PK and a master secret key MSK.

KeyGen$(MSK, PK, X) \to SK_X$: The key generation algorithm takes as input the master secret key MSK, the public key PK, and a key attribute $X \in A_n$. It outputs a private key SK_X. We assume X is included in SK_X implicitly.

Encrypt$(PK, \mathsf{M}, Y) \to CT$: The encryption algorithm takes as input a public key PK, the message M, and a ciphertext attribute $Y \in B_n$. It will output a ciphertext CT.

Decrypt$(PK, CT, Y, SK_X) \to \mathsf{M}$ or $\perp$: We assume that the decryption algorithm is deterministic. The decryption algorithm takes as input the public parameters PK, a ciphertext CT, ciphertext attribute $Y \in B_n$ and a private key SK_X. It outputs the message M or $\perp$ which represents that the ciphertext is not in a valid form. We require that the decryption algorithm outputs $\perp$ if $R(X, Y) = 0$.

We require correctness of decryption: that is, for all λ, all n, all $(PK, MSK) \in$ [**Setup**(λ, n)], all $X \in A_n, Y \in B_n$ such that $R(X, Y) = 1$, all $CT \in$ [**Encrypt** (PK, M, Y)] and all $SK_X \in$ [**KeyGen**(MSK, PK, X)], **Decrypt**$(PK, CT, Y, SK_X) = \mathsf{M}$ holds.

Security. We now define the security notion indistinguishability under chosen ciphertext attack (CCA-security) for an PE scheme Π. This security notion is defined by the following game between a challenger and attacker $\mathcal{A}$.

At first, the challenger runs the setup algorithm and gives PK to $\mathcal{A}$. Then $\mathcal{A}$ may adaptively make key-extraction queries and decryption queries. We denote this phase **Phase1**. In this phase, if $\mathcal{A}$ submits X to the challenger, the challenger returns $SK_X \leftarrow$ **KeyGen**(MSK, PK, X) if X has not been submitted before. Otherwise, the challenger returns the previously extracted SK_X. If $\mathcal{A}$ submits (CT, Y, X) to the challenger in a decryption query, the challenger extracts the private key for X by $SK_X \leftarrow$ **KeyGen**(MSK, PK, X) if this has not been previously extracted and returns the output of **Decrypt**(PK, CT, Y, SK_X) to $\mathcal{A}$. At some point, $\mathcal{A}$ outputs two equal length messages M_0 and M_1 and challenge ciphertext attribute $Y^\star \in B_n$. $Y^\star$ cannot satisfy $R(X, Y^\star) = 1$ for any attribute sets X such that $\mathcal{A}$ already queried private key for X. Then the challenger flips a random coin $\beta \in \{0, 1\}$, runs **Encrypt**$(PK, \mathsf{M}_\beta, Y^\star) \to CT^\star$ and gives challenge ciphertext $CT^\star$ to $\mathcal{A}$. In **Phase2**, $\mathcal{A}$ may adaptively make the same queries as in **Phase1** with following added restriction: $\mathcal{A}$ cannot make a key-extraction query for X such that $R(X, Y^\star) = 1$, and $\mathcal{A}$ cannot submit (CT, Y, X) such that $R(X, Y^\star) = 1$ and $(CT, Y) = (CT^\star, Y^\star)$. At last, $\mathcal{A}$ outputs a guess β' for β. We say that $\mathcal{A}$ succeeds if $\beta' = \beta$ and denote the probability of this event by $\Pr^{PE}_{\mathcal{A},\Pi}$. The advantage of an attacker $\mathcal{A}$ is defined as $Adv^{PE}_{\mathcal{A},\Pi} = \Pr^{PE}_{\mathcal{A},\Pi} - \frac{1}{2}$.

Definition 1. *We say that an PE scheme Π is* CCA-secure *(payload hiding)*[†] *if for all PPT $\mathcal{A}$, $Adv^{PE}_{\mathcal{A},\Pi}$ is negligible. We also say that an PE scheme Π is* CPA-secure *if for all PPT $\mathcal{A}$ who does not make any decryption queries, $Adv^{PE}_{\mathcal{A},\Pi}$ is negligible.*

We say that the PE scheme is selectively CCA/CPA-secure if we add an Initial stage **Init** before the setup where the adversary submits the target ciphertext attribute $Y^\star \in B_n$.

Typical Relations. An PE scheme captures the functionality of a large number of existing types of encryption schemes. In the following, we briefly illustrate how the most popular schemes can be obtained from an PE scheme by choosing the relation appropriately.

Broadcast Encryption. Broadcast encryption allows a sender to encrypt a message for any subset S of n users. To achieve this functionality, we set n to be the number of user, $A_n = \{1, 2, \ldots n\}$, $B_n = 2^{\{1,2,\ldots n\}}$. We define $R_n(j, S) = 1$ if and only if $j \in S$ for $j \in A_n$, $S \in B_n$.

Inner Product Encryption (for Non-Zero Relation). Inner product encryption (resp. for non-zero relation) allows a sender to encrypt a message for a vector $\boldsymbol{Y}$ so that a user with a secret key for a vector $\boldsymbol{X}$, can decrypt it if and only if $\boldsymbol{X} \cdot \boldsymbol{Y} = 0$ (resp. $\boldsymbol{X} \cdot \boldsymbol{Y} \neq 0$). To achieve this functionality, we set n to be dimension of the vectors, $A_n = \mathbb{Z}_N^n$, and $B_n = \mathbb{Z}_N^n$ where N is some integer determined by the scheme. We define $R_n(\boldsymbol{X}, \boldsymbol{Y}) = 1$ if and only $\boldsymbol{X} \cdot \boldsymbol{Y} = 0$ (resp. $\boldsymbol{X} \cdot \boldsymbol{Y} \neq 0$) for $\boldsymbol{X} \in A_n$, $\boldsymbol{Y} \in B_n$.

Spatial Encryption. Spatial encryption allows a sender to encrypt a message for some vector $\boldsymbol{V}$ so that a user with secret key for a space V such that $\boldsymbol{Y} \in V$ can decrypt it. To achieve this functionality, we set n to be the dimension of the vector, $A_n = \{\mathsf{Aff}(M, \boldsymbol{a}) | M \in \mathbb{Z}_N^{n \times l}, 0 \leq l \leq n, \boldsymbol{a} \in \mathbb{Z}_N^n\}$, $B_n = \mathbb{Z}_N^n$ where $\mathsf{Aff}(M, \boldsymbol{a}) = \{M\boldsymbol{x}^\top + \boldsymbol{a}^\top | \boldsymbol{x} \in \mathbb{Z}_N^l\}$ which is subspace of $\mathbb{Z}_N^n$. We define $R_n(V, \boldsymbol{Y}) = 1$ if and only if $\boldsymbol{Y}^\top \in V$ for $V \in A_n$, $\boldsymbol{Y} \in B_n$.

Key (Ciphertext) Policy Attribute based Encryption. Key (resp. ciphertext) policy attribute based encryption allows a sender to encrypt a message for some set of attribute S (resp. access structure $\mathbb{A}$) so that a user with secret key for an access structure $\mathbb{A}$ (resp. set of attribute S) such that $S \in \mathbb{A}$ can decrypt it. To achieve this functionality, we set n to be the size of attribute universe U. A_n is the collection of access structures over U (resp. $A_n = 2^U$). B_n is set as $B_n = 2^U$ (resp. access structure over U). Here, access structure over U can be described by linear secret sharing (LSSS) matrix whose size is bounded by some polynomial. We define $R_n(\mathbb{A}, S) = 1$ if and only if $\mathbb{A} \in A_n$ accepts $S \in B_n$.

[†] In this paper, we work only on payload-hiding security and not attribute-hiding which is considered for many other predicate encryption schemes such as [20].

2.2 Definition of Verifiability

In this subsection, we define verifiability of an PE scheme. Intuitively, verifiability guarantees that the decryption of any ciphertext is the same regardless of which user decrypt it, as long as this user is authorized to decrypt.

Definition 2. (Verifiablity) *An PE scheme Π is said to have* verifiability *if for all λ, n, $(PK, MSK) \in [\mathbf{Setup}(\lambda, n)]$, $X, X' \in A_n$, $Y \in B_n$ the following holds.*
If $SK_X \in [\mathbf{KeyGen}(MSK, PK, X)]$, $SK_{X'} \in [\mathbf{KeyGen}(MSK, PK, X')]$, and $R(X, Y) = R(X', Y)$, then for all $CT \in \{0, 1\}^*$, $\mathbf{Decrypt}(PK, CT, Y, SK_X) = \mathbf{Decrypt}(PK, CT, Y, SK_{X'})$ holds.

We remark that verifiability is not implied by correctness, since the definition of correctness is only concerned about correctly generated ciphertext whereas the definition of verifiability needs is about any ciphertext (including invalid one).

We also define public verifiability which is stronger notion than verifiability. That is, we can convert any PE scheme with public verifiability into PE scheme with verifiability very easily as we explain later. The reason why we introduce the notion of public verifiability is that in many case, we can check whether an PE scheme have public verifiability or not very easily.

Definition 3. (Public Verifiablity) *An PE scheme Π is said to have* public verifiability *if there exists a polynomial-time algorithm* **Verify** *which takes as input the public key PK, a possible ciphertext $CT \in \{0, 1\}^*$, a ciphertext attribute $Y \in B_n$ and outputs 0 or 1. We require that for all λ, n, $(PK, MSK) \in [\mathbf{Setup}(\lambda, n)]$, $Y \in B_n$, $CT \in \{0, 1\}^*$,*

$$\mathbf{Verify}(PK, CT, Y) = 1 \Leftrightarrow \exists \mathsf{M} \text{ such that } CT \in [\mathbf{Encrypt}(PK, \mathsf{M}, Y)].$$

An PE scheme with public verifiability can be modified to be verifiable by changing decryption algorithm slightly. That is, modified decryption algorithm $\mathbf{Decrypt}'(PK, CT, Y, SK_X)$ first checks whether $\mathbf{Verify}(PK, CT, Y) = 1$ holds and outputs $\mathbf{Decrypt}(PK, CT, Y, SK_X)$ if it holds. Otherwise it outputs $\perp$.

3 CCA-secure VPE from CPA-secure VPE

In this section, we show that VPE for a large class of relations can be transformed to be CCA-secure VPE with the same relation. Our requirement for this transformation is very weak, and many important relations defined for PE schemes satisfy this requirement. Our conversion works for wide class of PE such as spatial encryption, IPE, BE, KP/CP-ABE and can be seen as a nontrivial generalization of the conversion proposed by [37] which only works for ABE. We also remark that our conversion works for BE and IPE for which the Canetti-Halevi-Katz [12] transform cannot be applied in a straightforward manner.

3.1 Definitions

We define the notions of "OR-compatibility" and "equality test" for a relation. Intuitively, a relation is said to have OR-compatibility if for two attributes, the relation is able to capture the presence of one *or* the other, whereas a relation is said to support equality test over a domain D if it can be used to emulate an equality test for elements in D. The formal definitions are as follows:

Definition 4. (OR-COMPATIBILITY) *Consider a relation family* $R = \{R_n : A_n \times B_n \to \{0,1\} \mid n \in \mathbb{N}\}$. *We say that* R *is OR-compatible if for all* $n, m \in \mathbb{N}$ *there are maps* $\mathsf{OR} : B_n \times B_m \to B_{n+m}$ *and* $s : A_n \to A_{n+m}$ *and* $t : A_m \to A_{n+m}$ *such that for all* $X_1 \in A_n, X_2 \in A_m$ *and* $Y_1 \in B_n, Y_2 \in B_m$ *it holds that*

$$R_{n+m}(s(X_1), \mathsf{OR}(Y_1, Y_2)) = R_n(X_1, Y_1), R_{n+m}(t(X_2), \mathsf{OR}(Y_1, Y_2)) = R_m(X_2, Y_2).$$

Definition 5. (EQUALITY TEST) *Consider a relation family* $R = \{R_n : A_n \times B_n \to \{0,1\} \mid n \in \mathbb{N}\}$. *Consider a set* D. *We say that* R *can perform equality test over* D *by using dimension* d *if there are maps* $u : D \to A_d$ *and* $v : D \to B_d$ *such that for all* $z, z' \in D$ *we have* $R_d(u(z), v(z)) = 1$ *and* $R_d(u(z), v(z')) = 0$ *if* $z \neq z'$.

3.2 Generic Conversion

Let $\Pi = (\mathbf{Setup}, \mathbf{KeyGen}, \mathbf{Encrypt}, \mathbf{Decrypt})$ be a CPA-secure PE for relation R and let $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ be a strongly unforgeable one-time signature scheme. Here, $\mathcal{G}$, $\mathcal{S}$, and $\mathcal{V}$ are the key generation, sign, and verify algorithms of the scheme, respectively. Assume that Π has verifiability, OR-compatibility (as per definition 4), and can perform equality test (as per definition 5) over the verification key space of Σ. We can construct a CCA-secure VPE scheme $\Pi' = (\mathbf{Setup}', \mathbf{KeyGen}', \mathbf{Encrypt}', \mathbf{Decrypt}')$ also for relation R as follows.

Setup$'(\lambda, n)$**.** Output $\mathbf{Setup}(\lambda, n+d) \to (PK, MSK)$.
KeyGen$'(MSK, PK, X)$**.** Output $\mathbf{KeyGen}(MSK, PK, s(X)) \to SK_{s(X)}$. Hence $SK'_X = SK_{s(X)}$.
Encrypt$'(PK, \mathsf{M}, Y)$ First create a one-time signature key pair by running $\mathcal{G}(\lambda) \to (vk, sk)$. Then run $\mathbf{Encrypt}(PK, \mathsf{M}, \mathsf{OR}(Y, v(vk))) \to CT$ and $\mathcal{S}(sk, (CT, Y)) \to \sigma$. Lastly, output $CT' = (vk, CT, \sigma)$.
Decrypt$'(PK, CT', Y, SK'_X)$ Parses the ciphertext CT' as (vk, CT, σ). If $\mathcal{V}(vk, (CT, Y), \sigma) = 0$, output $\perp$. Output $\mathbf{Decrypt}(PK, CT, \mathsf{OR}(Y, v(vk)), SK'_X)$ otherwise.

CORRECTNESS. Decryption can be done using $SK_{s(X)}$ if $R_n(X, Y) = 1$ since $R_{n+d}(s(X), \mathsf{OR}(Y, v(vk))) = R_n(X, Y) = 1$. Thus, correctness of Π' follows from correctness of Π.

VERIFIABILITY. The verifiability of Π' follows directly from the verifiability of Π.

SELECTIVE SECURITY. Our conversion can be also applied to selectively (CPA-)secure PE schemes, and in such cases, resulting CCA-secure schemes are only selectively (CCA-)secure as well.

Theorem 1. *If Π is CPA-secure PE for relation R, then Π' is CCA-secure PE for relation R.*

Proof. Assume we are given PPT adversary $\mathcal{A}$ which breaks CCA-security of the scheme Π' for relation R_n with advantage ϵ. Then we construct another adversary $\mathcal{B}$ which breaks CPA-security of the scheme Π for relation R_{n+d} with advantage negligibly close to ϵ using $\mathcal{A}$. Define adversary $\mathcal{B}$ as follows:

Setup. The challenger runs $\mathbf{Setup}(\lambda, n+d) \rightarrow (PK, MSK)$. Then $\mathcal{B}$ is given PK and gives it to $\mathcal{A}$. $\mathcal{B}$ also runs $\mathcal{G}(\lambda) \rightarrow (vk^\star, sk^\star)$.

Phase1. $\mathcal{A}$ may adaptively make queries of the following types:

– ***Key-extraction query.*** When $\mathcal{A}$ submits X, then $\mathcal{B}$ submits $s(X)$ to challenger. $\mathcal{B}$ is given private key $SK_{s(X)}$ and gives it to $\mathcal{A}$.

– ***Decryption query.*** When $\mathcal{A}$ submits (CT', Y, X) such that $CT' = (vk, CT, \sigma)$, $\mathcal{B}$ respond to $\mathcal{A}$ as follows. First, $\mathcal{B}$ checks whether $R(X, Y) = 0$ or not. If so, $\mathcal{B}$ outputs $\perp$. Otherwise $\mathcal{B}$ checks whether $\mathcal{V}(vk, (CT, Y), \sigma) = 1$ holds. If it does not hold, then $\mathcal{B}$ returns $\perp$. If it holds and $vk^\star = vk$, then $\mathcal{B}$ aborts. Otherwise $\mathcal{B}$ submits $t(u(vk))$ to the challenger and is given $SK_{t(u(vk))}$. Then $\mathcal{B}$ returns output of $\mathbf{Decrypt}(PK, CT, \mathsf{OR}(Y, v(vk)), SK_{t(u(vk))})$ to $\mathcal{A}$.

Challenge. $\mathcal{A}$ declares two equal length messages M_0 and M_1 and an challenge attribute $Y^\star$. Then $\mathcal{B}$ declares the same messages M_0, M_1 and $\mathsf{OR}(Y^\star, v(vk^\star))$ for the challenger. The challenger flips a random coin $\beta \in \{0, 1\}$, runs $\mathbf{Encrypt}(PK, \mathsf{M}_\beta, \mathsf{OR}(Y^\star, v(vk^\star))) \rightarrow CT^\star$ and gives $CT^\star$ to $\mathcal{B}$. Then $\mathcal{B}$ runs $\mathcal{S}(sk^\star, (Y^\star, CT^\star)) \rightarrow \sigma^\star$, and gives $CT^{\star\prime} = (vk^\star, CT^\star, \sigma^\star)$ to $\mathcal{A}$.

Phase2. $\mathcal{B}$ responds to $\mathcal{A}$'s queries as the same as in **Phase1**.

Guess. Finally, $\mathcal{A}$ outputs a guess β' for β. Then $\mathcal{B}$ outputs β' as its guess.

First we check that the key extraction query of $\mathcal{A}$ is legal. $\mathcal{B}$ can submit $s(X)$ to the challenger, since $R_{n+d}(s(X), \mathsf{OR}(Y^\star, v(vk^\star))) = R_n(X, Y^\star) = 0$. $\mathcal{B}$ can also submits $t(u(vk))$ to the challenger since $R_{n+d}(t(u(vk)), \mathsf{OR}(Y^\star, v(vk^\star))) = R_d(u(vk), v(vk^\star)) = 0$ if $vk \neq vk^\star$. Next, we see that in the simulation of decryption oracle, $\mathbf{Decrypt}(PK, CT, \mathsf{OR}(Y, v(vk)), SK_{t(u(vk))}) = \mathbf{Decrypt}(PK, CT, \mathsf{OR}(Y, v(vk)), SK_{s(X)})$ by the verifiability since $R_{n+d}(t(u(vk)), \mathsf{OR}(Y, v(vk))) = R_{n+d}(s(X), \mathsf{OR}(Y, v(vk)) = 1$ if $R(X, Y) = 1$. Thus the simulation is perfect if $\mathcal{B}$ does not abort.

Let **Win** denote the event that $\mathcal{A}$ correctly guess β, **Abort** denote the event that $\mathcal{B}$ aborts. If **Abort** does not occur, $\mathcal{B}$'s simulation is perfect. So, $\mathcal{B}$'s advantage for guessing β is estimated as $\Pr[\mathcal{B}$ correctly guesses $\beta] - \frac{1}{2} = \Pr[\mathbf{Win}|\overline{\mathbf{Abort}}]\Pr[\overline{\mathbf{Abort}}] - \frac{1}{2} \geq \Pr[\mathbf{Win}] - \Pr[\mathbf{Abort}] - \frac{1}{2} \geq \epsilon - \Pr[\mathbf{Abort}]$. Since $\Pr[\mathbf{Abort}] = negl(\lambda)$ due to the unforgeability of the one-time-signature, the proof is completed.

3.3 Qualifying Relations

In this subsection, we show that important relations defined for PE schemes in the literature satisfy OR-compatibility and can perform equality test by describing their corresponding maps $\mathsf{OR} : B_n \times B_m \to B_{n+m}, s : A_n \to A_{n+m}, t : A_m \to A_{n+m}, u : D \to A_d, v : D \to B_d$.

Inner Product Relation. Equality test can be performed with $d = 2$.

$$\mathsf{OR}(\boldsymbol{Y}_1, \boldsymbol{Y}_2) = \boldsymbol{Y}_1 || \boldsymbol{Y}_2, \quad s(\boldsymbol{X}) = \boldsymbol{X} || \mathbf{0}, \quad t(\boldsymbol{X}) = \mathbf{0} || \boldsymbol{X},$$
$$u(z) = (z, 1), \quad v(z) = (-1, z)$$

Spatial Relation. Equality test can be performed with $d = 1$.

$$\mathsf{OR}(\boldsymbol{Y}_1, \boldsymbol{Y}_2) = \boldsymbol{Y}_1 || \boldsymbol{Y}_2, \quad s(\mathsf{Aff}(M, \boldsymbol{a})) = \mathsf{Aff}(\begin{bmatrix} M & 0 \\ 0 & I_m \end{bmatrix}, \boldsymbol{a} || \mathbf{0}),$$
$$t(\mathsf{Aff}(M, \boldsymbol{a})) = \mathsf{Aff}(\begin{bmatrix} I_n & 0 \\ 0 & M \end{bmatrix}, \mathbf{0} || \boldsymbol{a}), \quad u(z) = \mathsf{Aff}(0, (z)), \quad v(z) = (z)$$

Here, I_m and I_n are unit matrices of size m and n respectively.

We can also capture the case of CP/KP-ABE by a technique in [37]. We need to generalize the definition of equality test to instantiate BE and non-zero IPE in our framework. See the full version for the details.

4 Anonymous Deniable Predicate Authentication

In this section, we introduce the notion of ADPA. Intuitively, ADPA is a generalization of deniable authentication in which the prover holds a private key corresponding to an attribute, and the verifier will learn nothing about this attribute, except that it satisfies a relation with the verifier attribute. Firstly, we define functionality and security, and then show how a ADPA scheme can be constructed from a CCA-secure VPE scheme.

4.1 Definition of Anonymous Deniable Predicate Authentication

Syntax. Let relation $R = \{R_n : A_n \times B_n \to \{0, 1\} \mid n \in \mathbb{N}\}$ be a collection of boolean functions, where $n \in \mathbb{N}$ denotes a "scheme description", A_n and B_n denote the "prover attribute" and "verifier attribute" spaces. An *anonymous deniable predicate authentication* (ADPA) for a relation R is defined by a tuple of four algorithms $\Lambda_R^{\text{DPA}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{P}, \mathsf{V})$. The setup algorithm Setup takes as inputs a security parameter λ, a scheme description $n \in \mathbb{N}$, and outputs a public key PK and a master secret key MSK. And the key generation algorithm KeyGen takes as inputs the master secret key MSK, the public key PK, a prover attribute $X \in A_n$, and outputs a private key SK_X. The interactive Turing machines prover P and verifier V perform the interactive protocol (P, V) with common inputs PK, $Y \in B_n$, message M, where P also takes input the private key for X, SK_X. At the end of protocol (P, V) , V outputs a bit to indicate whether V accepts P as a valid prover or not. With $\langle \mathsf{P}(x_\mathsf{P}), \mathsf{V}(x_\mathsf{V}) \rangle (y)$, we denote the output of verifier V at the end of execution of interactive protocol (P, V), where P and V take x_P and x_V as private inputs respectively and y denotes the common input. For the basic requirement *completeness*, Λ_R^{DPA} needs to satisfy that for all λ, $n \in \mathbb{N}$, $X \in A_n$, $Y \in B_n$ such that $R_n(X,Y)=1$, M, the following holds.

$$(PK, MSK) \leftarrow \mathsf{Setup}(\lambda, n), SK_X \leftarrow \mathsf{KeyGen}(MSK, PK, X)\ :\ \langle \mathsf{P}(SK_X), \mathsf{V}\rangle(PK, Y, \mathsf{M}) = 1.$$

For security, Λ_R^{DPA} is also required to satisfy the following notions.

Concurrent Soundness. First, we define the adversary $\mathcal{A}$ as a *man-in-the-middle* attacker such that $\mathcal{A}$ is interacting with provers $\mathsf{P}_1, \ldots, \mathsf{P}_{m_L}$ in m_L "left sessions" as verifier, and at the same time interacting with an honest verifier V in a "right session" as prover, in any arbitrary interleaving, where m_L is polynomial in security parameter λ. The adversary $\mathcal{A}$ is given access to two additional oracles: (1) prover instantiator oracle $\mathcal{P}$, and (2) key generator oracle $\mathcal{K}$.

When $\mathcal{A}$ submits to prover instantiator oracle $\mathcal{P}$ a message M, a verifier attribute Y, and a prover attribute X, $\mathcal{P}$ will initiate a new prover P' with inputs $(PK, Y, \mathsf{M}, SK_X)$, where SK_X is a valid secret key corresponding to the key attribute X. The adversary $\mathcal{A}$ is allowed to send a prover attribute X and then retrieve the corresponding secret key SK_X from the key generator oracle $\mathcal{K}$ with the restriction that $R(X, Y^\star) \neq 1$ holds, where $Y^\star$ is the verifier attribute used as common input in the right session. The following notion guarantees that such adversary $\mathcal{A}$ will not be able to make the honest verifier V to accept it as a valid prover in right session.

Definition 6 (Concurrent Soundness). *Let* $\Lambda_R^{\mathrm{DPA}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{P}, \mathsf{V})$ *be an ADPA for relation* $R = \{R_n : A_n \times B_n \to \{0,1\} \mid n \in \mathbb{N}\}$*. We say that* Λ_R^{DPA} *satisfies concurrent soundness if for all sufficiently large* λ*, for any* $n \in \mathbb{N}$*, for any efficient algorithm* $\mathcal{A}$*, the following holds.*

$$\Pr\left[\mathsf{Setup}(\lambda, n) \to (PK, MSK)\ :\ \substack{\mathcal{A}^{\mathcal{P},\mathcal{K}}(PK) \to (\mathsf{M}^\star, Y^\star, state) \\ \langle \mathcal{A}^{\mathcal{P},\mathcal{K}}(state), \mathsf{V}\rangle(PK, Y^\star, \mathsf{M}^\star) = 1}\right] = negl(\lambda),$$

where

- *the key generator oracle* $\mathcal{K}$*, on input a prover attribute* $X_i \in A_n$ *such that* $R_n(X_i, Y^\star) \neq 1$ *holds, returns* $SK_{X_i} \leftarrow \mathsf{KeyGen}(PK, MSK, X_i)$*,*
- *the prover instantiator oracle* $\mathcal{P}$*, on input a tuple* $(\mathsf{M}_i, Y_i \in B_n, X_i \in A_n)$ *such that* $(\mathsf{M}_i, Y_i) \neq (\mathsf{M}^\star, Y^\star)$ *holds, allows* $\mathcal{A}$ *access to a prover* $\mathsf{P}_i \in \{\mathsf{P}_1, \ldots, \mathsf{P}_{m_L}\}$ *which has been initiated with inputs* $(PK, Y_i, \mathsf{M}_i, SK_{X_i})$*, where* $SK_{X_i} \leftarrow \mathsf{KeyGen}(PK, MSK, X_i)$*,*
- $\mathcal{A}$ *interacts as a verifier with provers* $\mathsf{P}_1, \ldots, \mathsf{P}_{m_L}$ *generated by* $\mathcal{P}$ *concurrently, and for each instantiated prover* $\mathsf{P}_1, \ldots, \mathsf{P}_{m_L}$*, and* $\mathcal{A}$ *interacts in the protocol* $(\mathsf{P}_i, \mathcal{A})$ *with common inputs* PK*,* $Y_i \in B_n$*,* M*, where* P_i *also takes input* SK_{X_i}*.*

We remark that we can consider a weaker version of the above security notion in which the adversary is required to output $Y^\star$ at the beginning of the game. We denote this security selective concurrent soundness. Next, we define the special security notions we require an ADPA to satisfy.

Anonymity (Source Hiding). Here we describe the security notion which guarantees that no one is able to know which key attribute is associated to the prover P in the interactive protocol (P, V), even when one is allowed to act as a *cheating* verifier in the interactive protocol and is given access to the master secret key generated by the setup algorithm.

Let $R = \{R_n : A_n \times B_n \to \{0,1\} \mid n \in \mathbb{N}\}$ be a relation and $\Lambda_R^{\text{DPA}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{P}, \mathsf{V})$ be an ADPA for R. Let us consider an adversary $\mathcal{A}$ which engages in the following game.

$\mathsf{Game}_{\mathcal{A}}^{\text{anom}}(\lambda, n)$:

$$\begin{aligned}
&\mathsf{Setup} \to (PK, MSK)\ ,\ \mathcal{A}(PK, MSK) \to (X_0^\star, X_1^\star)\\
&\mathsf{KeyGen}(MSK, PK, X_0^\star) \to SK_{X_0^\star}\ ,\ \mathsf{KeyGen}(MSK, PK, X_1^\star) \to SK_{X_1^\star}\\
&\mathcal{A}(PK, MSK) \to (Y^\star, \mathsf{M}^\star, \mathit{state}) \text{ s.t. } R_n(X_0^\star, Y^\star) = R_n(X_1^\star, Y^\star)\\
&b \xleftarrow{\$} \{0,1\}\ ,\ \mathcal{A}^{\mathsf{P}(SK_{X_b^\star})}(\mathit{state}, SK_{X_0^\star}, SK_{X_1^\star}, MSK, PK, Y^\star, \mathsf{M}^\star) \to \hat{b}\\
&\text{If } b = \hat{b} \text{ return 1, otherwise return 0.}
\end{aligned}$$

The notation $\mathcal{A}^{\mathsf{P}(SK_{X_b^\star})}(\mathit{state}, SK_{X_0^\star}, SK_{X_1^\star}, MSK, PK, Y^\star, \mathsf{M}^\star)$ in $\mathsf{Game}_{\mathcal{A}}^{\text{anom}}(\lambda, n)$ denotes that $\mathcal{A}$ interacts as verifier with P in the interactive protocol $(\mathsf{P}, \mathcal{A})$ with common inputs $(PK, Y^\star, \mathsf{M}^\star)$ where P also takes input secret key $SK_{X_b^\star}$ and $\mathcal{A}$ also takes inputs $(\mathit{state}, SK_{X_0^\star}, SK_{X_1^\star}, MSK)$. The following notion guarantees that there no such adversary $\mathcal{A}$ will be able to correctly guess whether P uses $SK_{X_0^\star}$ or $SK_{X_1^\star}$ as its secret key.

Definition 7 (Anonymity (Source Hiding)). *Let* $\Lambda_R^{\text{DPA}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{P}, \mathsf{V})$ *be an ADPA for relation* $R = \{R_n : A_n \times B_n \to \{0,1\} \mid n \in \mathbb{N}\}$*. We say that* Λ_R^{DPA} *satisfies anonymity (source hiding) if for all sufficiently large* λ*, for any* $n \in \mathbb{N}$*, for any machine* $\mathcal{A}$ *with unbounded power,* $\left|\Pr\left[\mathsf{Game}_{\mathcal{A}}^{\text{anom}}(\lambda, n) = 1\right] - \frac{1}{2}\right| = \mathit{negl}(\lambda)$ *holds.*

Deniability. Here we describe the security notion which guarantees that the communication transcript which is produced from an interaction between prover and verifier in an ADPA cannot be used as a proof that an interaction between prover and verifier has taken place. More precisely, the security notion says that for any verifier (including dishonest verifiers), there exists a simulator which can poses as a valid prover even without knowledge about the secret key. Also, with $\mathsf{View}(\langle P, V\rangle)$, we denote the view which is obtained at the end of interaction between P and V, where the view is the communication transcripts concatenated by random coins used by V.

Definition 8 (Deniability). *Let* $R = \{R_n : A_n \times B_n \to \{0,1\} \mid n \in \mathbb{N}\}$ *be a relation and* $\Lambda_R^{\text{DPA}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{P}, \mathsf{V})$ *be an ADPA for relation* R*. Let us also define the following two probability distributions for a fixed* λ*,* $n \in \mathbb{N}$*,* M*,* *and $X \in A_n$, $Y \in B_n$ such that $R_n(X,Y) = 1$.*

$$Real(\lambda, n, X, Y, \mathsf{M}) = \begin{bmatrix} \mathsf{Setup}(\lambda, n) \to (PK, MSK), \\ \mathsf{KeyGen}(PK, MSK, X) \to SK_X, \\ \mathsf{View}\left(\langle \mathsf{P}(SK_X), \mathcal{A}(MSK, X)\rangle (PK, Y, \mathsf{M})\right) \end{bmatrix},$$

$$Sim(\lambda, n, X, Y, \mathsf{M}) = \begin{bmatrix} \mathsf{Setup}(\lambda, n) \to (PK, MSK), \\ \mathsf{KeyGen}(PK, MSK, X) \to SK_X, \\ \mathsf{View}\left(\langle \mathsf{Sim}, \mathcal{A}(MSK, X)\rangle (PK, Y, \mathsf{M})\right) \end{bmatrix},$$

where $\mathcal{A}$ and Sim are both efficient algorithms. Λ_R^{DPA} is said to be deniable if for all sufficiently large λ, for any $n \in \mathbb{N}$, M, and for all $X \in A_n$, $Y \in B_n$ such that $R_n(X,Y) = 1$, the following holds.

$$\forall \mathcal{A}\ \exists \mathsf{Sim} : Real(\lambda, n, X, Y, \mathsf{M}) \overset{c}{\approx} Sim(\lambda, n, X, Y, \mathsf{M}).$$

4.2 Construction from CCA-secure VPE

We can construct an ADPA $\Lambda_R^{\mathrm{DPA}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{P}, \mathsf{V})$ from a CCA secure VPE $\Pi = (\mathbf{Setup}, \mathbf{KeyGen}, \mathbf{Encrypt}, \mathbf{Decrypt})$ and a perfectly binding and computationally hiding commitment scheme $COM = (\mathsf{com}, \mathsf{open})$. Here, com and open are commit and open algorithms of the scheme, respectively. Setup and KeyGen are exactly the same as **Setup** and **KeyGen**. We describe the interactive protocol (P, V) as follows. Note that P and V perform (P, V) with common input the public key PK, a verifier attribute $Y \in B_n$, and a message M, while P also takes as input the secret key SK_X corresponding to a key attribute $X \in A_n$ such that $R_n(X,Y) = 1$ holds.

Step1 $(\mathsf{P} \Leftarrow \mathsf{V})$**:** V chooses randomly $r \leftarrow \{0,1\}^\lambda$ and then computes $CT \leftarrow \mathbf{Encrypt}(PK, \mathsf{M}||r, Y)$. Then V sends CT to P.

Step2 $(\mathsf{P} \Rightarrow \mathsf{V})$**:** P computes $y \leftarrow \mathbf{Decrypt}(PK, CT, Y, SK_X)$. If $y = \perp$ or $y = \mathsf{M}'||r'$ such that $\mathsf{M}' \neq \mathsf{M}$, P chooses random pairs $(r_{i0}, r_{i1}) \in \{0,1\}^\lambda \times \{0,1\}^\lambda$ for $i = 1, \ldots, \lambda$. Otherwise, P chooses $(r_{i0}, r_{i1}) \in \{0,1\}^\lambda \times \{0,1\}^\lambda$ such that $r_{i0} \oplus r_{i1} = r'$ holds for all $i = 1, \ldots \lambda$. Then P sends $\{(C_{i0}, C_{i1}) = (\mathsf{com}(\sigma_{i0}, r_{i0}), \mathsf{com}(\sigma_{i1}, r_{i1}))\}_{i=1,\ldots,\lambda}$ to V, where σ_{i0} and σ_{i1} are randomnesses used to calculate the commitments C_{i0} and C_{i1} respectively.

Step3 $(\mathsf{P} \Leftarrow \mathsf{V})$**:** V sends λ random bits $b_1, b_2, \ldots b_\lambda$ to P.

Step4 $(\mathsf{P} \Rightarrow \mathsf{V})$**:** P sends $\{r_{ib_i} = \mathsf{open}(\sigma_{ib_i}, C_{ib_i}), \sigma_{ib_i}\}_{i=1,2,\ldots,\lambda}$ to V.

Step5 $(\mathsf{P} \Leftarrow \mathsf{V})$**:** V opens CT by revealing r and ρ to P, where ρ is randomness used to create CT.

Step6 $(\mathsf{P} \Rightarrow \mathsf{V})$**:** P sends $\{r_{i\bar{b}_i} = \mathsf{open}(\sigma_{i\bar{b}_i}, C_{i\bar{b}_i}), \sigma_{i\bar{b}_i}\}_{i=1,2,\ldots,\lambda}$ to V.
V outputs 1 if for all $1 \leq i \leq \lambda$, $r_{i0} \oplus r_{i1} = r$, and outputs 0 otherwise.

We remark that our conversion can also be applied to selectively CCA-secure PE schemes, and in this case, the resulting ADPA schemes satisfies only selective concurrent soundness.

4.3 Security Analysis

Theorem 2. *If Π is CCA-secure VPE and COM is perfectly binding and computationally hiding commitment, then ADPA Λ_R^{DPA} constructed as above satisfies concurrent soundness, deniability, and anonymity. Especially, source hiding is satisfied for any adversary (even unbounded).*

The theorem can be proved following a very similar strategy to that of [15, 25], and we will therefore only sketch how the proof is obtained in the following.

Lemma 1. *(Concurrent Soundness.) If Π is CCA-secure PE and COM is perfectly secure binding commitment, then Λ_R^{DPA} satisfies concurrent soundness.*

Similar to the case of [25], it is easy to see that the above construction satisfies concurrent soundness. We can construct an IND-CCA adversary $\mathcal{B}$ of the underlying predicate encryption using adversary $\mathcal{A}$ who violates soundness of above scheme. Note that $\mathcal{B}$ can easily simulate the prover instantiator oracle $\mathcal{P}$ perfectly using the given decryption oracle and also simulate the key generator oracle $\mathcal{K}$ perfectly via key extraction queries. The key point of the proof is that $\mathcal{B}$ is allowed to rewind $\mathcal{A}$ and to let $\mathcal{A}$ answer two different sequences of $\{b_i\}_{i=1,\ldots,\lambda}$ in **Step3**, so that $\mathcal{B}$ is able to compute $r = r_{i0} \oplus r_{i1}$ for some $i \in [1, \lambda]$. In the challenge phase, $\mathcal{B}$ can select two messages M_0, M_1 such that $\mathsf{M}_0 = \mathsf{M}||\widehat{r}$, $\mathsf{M}_1 = \mathsf{M}||\widetilde{r}$. and then forward the received challenge ciphertext $CT^\star$ to $\mathcal{A}$ in **Step1**. Since $\mathcal{B}$ can obtain r from $\mathcal{A}$ through the rewinding described above, where $\mathsf{M}||r$ is the result of the decryption of $CT^\star$, $\mathcal{B}$ can check whether $\widehat{r} = r$ or $\widetilde{r} = r$, and thereby easily determine whether $CT^\star$ is the encryption of M_0 or M_1.

Lemma 2. *(Deniability.) If Π satisfies correctness and COM is computationally hiding commitment, then Λ_R^{DPA} satisfies deniability.*

One can prove that the above instantiation is deniable using the same techniques as shown in [15, 25]. Intuitively, the procedure to construct the simulator Sim is to firstly run the interaction with the verifier until **Step5** where the verifier has to reveal the randomness r it used in **Step1** to create the CT, and then rewind the verifier until the end of **Step1**. In the second run after the rewind, we can easily simulate a prover until the last step **Step6**, since the randomness r should have been obtained in the first run. (Here, we resort to correctness of Π.) The most crucial point here is how to safely perform **Step2** in the first run (before the rewind). The trick is that although we do not know the randomness r yet, we can send commitments of random messages to $\mathcal{A}$ in **Step2**, as the computationally hiding property of the underlying commitment COM prevents $\mathcal{A}$ from detecting that the commitments sent by Sim are actually commitments to random messages.

Lemma 3. *(Anonimity) If Π is VPE, then Λ_R^{DPA} satisfies anonymity.*

Anonymity of the scheme follows immediately from verifiability. Notice that the difference between an interaction with a prover which uses $SK_{X_0^\star}$ and an interaction with a prover which uses $SK_{X_1^\star}$ will only possibly occur at **Step2**, when the prover decrypts the ciphertext CT sent by the verifier at **Step1**. Thanks to the verifiable property of the underlying verifiable predicate encryption scheme, the result of the decryption is always the same, both in the case of $SK_{X_0^\star}$ and $SK_{X_1^\star}$, as long as $R_n(X_0^\star, Y) = R_n(X_1^\star, Y)$. Note that security level of anonymity achieved by our scheme is stronger than that of [25]. We achieve anonymity even against an adversary with unbounded computational power, whereas [25] only achieves anonymity against a computational bounded adversary.

5 Instantiations

To be able to apply our framework for constructing CCA-secure PE schemes or ADPA schemes, we require that the underlying PE schemes are verifiable. We note that many selectively-secure PE schemes [8, 9, 17, 29, 22, 3, 7, 16, 36] have public verifiability. That is, we can construct an **Verify** algorithm (as per definition 3). Hence, these can be used directly in our framework. On the other hand, this is not the case for the PE schemes with full security [35, 1, 2, 21, 28]. This is because all existing fully secure PE schemes make use of the dual system encryption methodology [35]. The security of these schemes rely on the indistinguishability between normal ciphertexts and semi-functional ciphertexts where a semi-functional ciphertext is special kind of incorrectly generated ciphertext. To achieve public verifiability, we should be able to distinguish between a semi-functional (i.e. incorrectly generated) ciphertext and a normal (i.e. correctly generated) ciphertext efficiently, but this conflicts with the security of the scheme. However, even though we cannot achieve public verifiability for these schemes, it is possible to achieve our definition of (non-public) verifiability. Recall that our definition of verifiability does not require that we can check whether ciphertext is correctly generated or not, but only requires that we can check whether the decryption of a ciphertext under a different secret key will be the same or not.

In the following, we first discuss how we add verifiability to the schemes in [35, 1, 2, 21, 28], then focus on the schemes which we obtain by applying our framework to the above mentioned verifiable PE schemes.

Modifying Existing Schemes to be Verifiable. Here, we explain how we modify the schemes in AL10 spatial encryption scheme [2], OT10 KP-ABE scheme [28], Waters09 BE scheme [35], AL10 IPE scheme [1] to be verifiable. Our first approach is to modify the original scheme so that its decryption algorithm first checks the validity of a ciphertext to a certain extent. We cannot check the validity of the ciphertext perfectly because of the above reason, but for the AL10 spatial encryption scheme and the OT10 KP-ABE scheme, this partial validity check is enough to prove verifiability. (We remark that in the modification, we also make some parts of master secret key public. The anonymity of OT10 scheme is lost by this modification.) For the Waters09 BE and AL10 IPE schemes, the above strategy does not seem to be enough. We then further modify these

Table 1. Overview of existing PE schemes. In the table. “PubVer” represents that the scheme has public verifiability. “Veri” represents that the schemet can be modified to be verifiable.

Schemes		Type	Verif.	Security	Assumption
Boneh et al.	[8, Sec. 3]	BE	PubVer	selective	D-l-BDHE
Boneh et al.	[8, Sec. 5]	BE	PubVer	selective	D-l-BDHE
Waters	[35, Sec. 5]	BE	Veri	full	DLIN and DBDH
Boneh et al.	[9]	Spatial	PubVer	selective	BDDHE
Attrapadung et al.	[2, Sec. B]	Spatial	Veri	full	3assumptions
Attrapadung et al.	[1]	IPE	Veri	full	DLIN and DBDH

schemes so that a user has some additional keys for the same attributes, but which uses different randomness. Then, in the decryption algorithm, the user checks whether the decryption of a ciphertext using different keys are the same or not. If it is different, then it indicates that the ciphertext is invalid. With this modification, we can prove verifiability of these schemes. For the description of the schemes and proofs of security and verifiability, see the full version of this paper.

CCA secure PE schemes. Since our conversion works for PE schemes which are not captured by the CPA-ABE to CCA-ABE conversion proposed by [37], we obtain a number of new CCA-secure PE schemes. Especially, our conversion works for BE, IPE, and spatial encryption scheme. In Tabel. 1, we list some candidate scheme in this category which we can use as underlying schemes in our framework. Hence, we can obtain an adaptively secure CCA-secure BE scheme by applying our conversion to the Waters09 BE scheme [35]. Furthermore, we also obtain a new selectively and adaptively secure CCA spatial encryption scheme by applying our conversion to the Boneh-Hamburg [9] and AL10 [2] spatial encryption schemes, respectively. Finally, we also obtain a new adaptively secure CCA IPE scheme by applying our conversion to [1]. We also note that it is easy to modify the Katz-Sahai-Waters [20] scheme to have verifiability. But the anonymity of the scheme is lost by this modification. Furthermore, it seems possible to transform the schemes [27] and [21] into verifiable variants. We note that our conversion also works for the ABE schemes [35, 1, 2, 21, 28], since our conversion capture the case of ABE as well. We also note that a special case of our conversion is considered in [18] in a context of BE. But they do not consider how to apply the conversion to the Waters09 BE scheme.

ADPA schemes. Our CCA-secure VPE scheme to ADPA scheme conversion works for all the schemes we obtained above. Hence, we can obtain a deniable ring authentication system with (adaptive) concurrent soundness and constant size ciphertexts by applying our conversion to the CCA-secure BE obtained above. As far as we know, this is the first time a scheme with these properties have been proposed. Furthermore, we can obtain an ADPA for a spatial relation and with selective and adaptive concurrent soundness by applying our conversion to the CCA-secure spatial encryption scheme obtained above. We can also obtain an ADPA for an inner product relation with (adaptive) concurrent soundness by applying our conversion to adaptively secure CCA IPE scheme obtained above. All of these schemes are new types of deniable authentication schemes. We can also see that if we use a CCA-secure ABE as the underlyingscheme (obtained by applying the transformation in [37] to [17, 29, 22, 3, 7, 16, 36]), then we obtain an ADPA for an attribute based relation.

References

1. N. Attrapadung, B. Libert. Functional Encryption for Inner Product: Achieving Constant-Size Ciphertexts with Adaptive Security or Support for Negation. In *PKC'10*, *LNCS* 6056, pp. 384–402. Springer, 2010.
2. N. Attrapadung, B. Libert. Functional Encryption for Public-Attribute Inner Products: Achieving Constant-Size Ciphertexts with Adaptive Security or Support for Negation. In *Journal of Mathematical Cryptology*, October 2011, vol. 5, n2, pp. 115-158. This is full version of [1].
3. N. Attrapadung, B. Libert. Expressive Key-Policy Attribute-Based Encryption with Constant-Size Ciphertexts. In *PKC'11*, *LNCS* 6571, pp. 17-34, 2011.
4. D. Boneh, X. Boyen. Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In *Eurocrypt'04*, *LNCS* 3027, pp. 223–238, 2004.
5. D. Boneh, M. Franklin. Identity-Based Encryption from the Weil Pairing. In *Crypto'01*, *LNCS* 2139, pp. 213–229, 2001.
6. M. Barbosa and P. Farshim. Delegatable Homomorphic Encryption with Applications to Secure Outsourcing of Computation. In *CT-RSA'12*, *LNCS* 7178, pp. 296-312, 2012.
7. J. Bethencourt, A. Sahai, B. Waters. Ciphertext-Policy Attribute-Based Encryption. IEEE Symposium on Security and Privacy (S&P), pp. 321-334, 2007.
8. D. Boneh, C. Gentry, B. Waters. Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys. In *Crypto'05*, *LNCS* 3621, pp. 258–275, 2005.
9. D. Boneh, M. Hamburg. Generalized Identity Based and Broadcast Encryption Schemes. In *Asiacrypt'08*, *LNCS* 5350, pp. 455–470, 2008.
10. D. Boneh, J. Katz.Improved Efficiency for CCA-Secure Cryptosystems Built Using Identity-Based Encryption. In *CT-RSA'05*, *LNCS* 3376, pp. 87-103, 2005.
11. D. Boneh, B. Waters. Conjunctive, Subset, and Range Queries on Encrypted Data. In *TCC'07*, *LNCS* 4392, pp535-554, 2007.
12. R. Canetti, S. Halevi, J. Katz. Chosen-Ciphertext Security from Identity-Based Encryption. In *Eurocrypt'04*, *LNCS* 3027, pp. 207-222, 2004.
13. C. Dwork, M. Naor, A. Sahai. Concurrent Zero-Knowledge. In Journal of the ACM (JACM), v.51 n.6, pp. 851-898, 2004. Preliminary version In *STOC'98*.
14. R. Gennaro. Multi-trapdoor Commitments and Their Applications to Proofs of Knowledge Secure Under Concurrent Man-in-the-Middle Attacks. In *Crypto'04*, *LNCS* 3152, pp. 220–236, 2004.
15. O. Goldreich, A. Kahan. How to Construct Constant-Round Zero-Knowledge Proof Systems for NP. In *Journal of Cryptology*, Volume 9, 167-189, 1996.
16. V. Goyal, A. Jain, O. Pandey, A. Sahai. Bounded Ciphertext Policy Attribute Based Encryption. ICALP (2) 2008, *LNCS* 5126, pp. 579–591, 2008.
17. V. Goyal, O. Pandey, A. Sahai, B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM CCS'06*, pp. 89–98, 2006.

18. G. Hanaoka and K. Kurosawa. Efficient Chosen Ciphertext Secure Public Key Encryption under the Computational Diffie-Hellman Assumption. In *Asiacrypt'08*, *LNCS* 5350, pp. 308-325, 2008. Full version is available at http://eprint.iacr.org/2008/211
19. J. Katz. Efficient and Non-malleable Proofs of Plaintext Knowledge and Applications. In *Eurocrypt 2003*, *LNCS* 2656, pp.211-228, 2010.
20. J. Katz, A. Sahai, B. Waters. Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products. In *Eurocrypt'08*, *LNCS* 4965, pp. 146-162, 2008.
21. A. Lewko, T. Okamoto, A. Sahai, K. Takashima, B. Waters. Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption. In *Eurocrypt '10*, *LNCS* 6110, pp.62-91, 2010.
22. A. Lewko, A. Sahai, B. Waters. Revocation Systems with Very Small Private Keys. In IEEE Symposium on Security and Privacy (S&P) pp.273-285, 2010.
23. A. Lewko and B.Waters, Efficient Pseudorandom Functions from the Decisional Linear Assumption and Weaker Variants. In *ACM-CCS'09*, pp112-120, 2009.
24. A. Lewko and B.Waters. New Techniques for Dual System Encryption and Fully Secure HIBE with Short Ciphertexts. In *TCC'10*, *LNCS* 5978, pp455-479, 2010.
25. M. Naor. Deniable Ring Authentication. In *Crypto'02*, *LNCS* 2442, pp. 481-498, 2010.
26. M. Naor, M. Yung. Public-key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. In *STOC'90* , pp427-437, 1990.
27. T. Okamoto, K. Takashima. Hierarchical Predicate Encryption for Inner-Products. In *Asiacrypt'09*, *LNCS* 5912, pp. 214-231, 2009.
28. T. Okamoto, K. Takashima. Fully secure functional encryption with general relations from the decisional linear assumption.In *Crypto'10*, *LNCS* 6223, pp. 191-208, 2010. Full version is available at http://eprint.iacr.org/2010/563.
29. R. Ostrovsky, A. Sahai, B. Waters. Attribute-based encryption with non-monotonic access structures. In *ACM CCS'07*, pp. 195–203, 2007.
30. R. Pass. On Deniability in the Common Reference String and Random Oracle Model. In *Crypto'03*, *LNCS* 2729, pp. 316-337, 2010.
31. M. D. Raimondo, R. Gennaro. New approaches for deniable authentication. In *ACM-CCS'05*, pp112-121, 2005.
32. R. Rivest, A. Shamir, Y. Tauman. How to leak a secret. In *Asiacrypt'01*, *LNCS* 2248, pp. 552-565, 2001.
33. A. Sahai, B. Waters. Fuzzy Identity-Based Encryption. In *Eurocrypt'05*, *LNCS* 3494, pp457-473, 2005.
34. B. Waters. Efficient Identity-Based Encryption Without Random Oracles. In *Eurocrypt'05*, *LNCS* 3494, pp. 114–127, 2005.
35. B. Waters. Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In *Crypto'09*, *LNCS* 5677, pp. 619–636, 2009. Full version is available at http://eprint.iacr.org/2009/385.
36. B. Waters. Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization. In *PKC'11*, *LNCS* 6571, pp. 53-70, 2011.
37. S. Yamada, N. Attrapadung, G. Hanaoka, and N. Kunihiro. Generic Constructions for Chosen-Ciphertext Secure Attribute Based Encryption. In *PKC'11*, *LNCS* 6571, pp. 71-89, 2011.