Multi-Designated Receiver Signed Public Key Encryption*

Ueli Maurer[1], Christopher Portmann[2], and Guilherme Rito[1]

[1] Department of Computer Science, ETH Zürich, Switzerland
{maurer,gteixeir}@inf.ethz.ch
[2] Concordium, Zürich, Switzerland
cp@concordium.com

Abstract. This paper introduces a new type of public-key encryption scheme, called Multi-Designated Receiver Signed Public Key Encryption (MDRS-PKE), which allows a sender to select a set of designated receivers and both encrypt and sign a message that only these receivers will be able to read and authenticate (*confidentiality* and *authenticity*). An MDRS-PKE scheme provides several additional security properties which allow for a fundamentally new type of communication not considered before. Namely, it satisfies *consistency*—a dishonest sender cannot make different receivers receive different messages—*off-the-record*—a dishonest receiver cannot convince a third party of what message was sent (e.g., by selling their secret key), because dishonest receivers have the ability to forge signatures—and *anonymity*—parties that are not in the set of designated receivers cannot identify who the sender and designated receivers are.

We give a construction of an MDRS-PKE scheme from standard assumptions. At the core of our construction lies yet another new type of public-key encryption scheme, which is of independent interest: Public Key Encryption for Broadcast (PKEBC) which provides all the security guarantees of MDRS-PKE schemes, except authenticity.

We note that MDRS-PKE schemes give strictly more guarantees than Multi-Designated Verifier Signature (MDVS) schemes with privacy of identities. This in particular means that our MDRS-PKE construction yields the first MDVS scheme with privacy of identities from standard assumptions. The only prior construction of such schemes was based on Verifiable Functional Encryption for general circuits (Damgård et al., TCC '20).

* This is the full version of article [33], ©IACR 2022, https://doi.org/10.1007/978-3-031-07085-3_22.

Erratum

The Off-The-Record security proof of the MDRS-PKE construction given in the prior full version of this paper [34] as well as in the published version [33] is wrong. To fix the issue, in this new version we modify the construction by adding a strongly unforgeable one-time signature scheme (that we use to bind MDVS signatures and PKEBC ciphertexts together) and update all the security proofs of the MDRS-PKE construction accordingly. We note that the security proof of our new construction relies on the unforgeability of the underlying MDVS; while the type of unforgeability we require the underlying MDVS scheme is stronger than the notion considered by Damgard et al. [16] (in particular, our notion provides adversaries with access to a signature verification oracle, which we use to handle decryption queries), the MDVS construction given in [12] does satisfy this new stronger notion (and is based on standard assumptions).

In this new version we also assume perfect correctness from the PKE scheme underlying the PKEBC construction, and updated the security analysis of our PKEBC construction accordingly.

While the two changes mentioned above are the main ones, there are other minor updates in this new full version.

1 Introduction

1.1 Public Key Encryption security properties

The most common use case for cryptography is sending a message to a single receiver. Here one usually desires to have *confidentiality* (only the desired receiver can read the message) and *authenticity* (the receiver is convinced that the message is from the declared sender). Although one might be interested in signatures that can be publicly verified (e.g. for a judge to verify a contract), when trying to protect the privacy of personal communication one often wants the opposite: not only is the intended receiver the only one that can verify the signature, but even if this person sells their secret key, no third party will be convinced of the authenticity of the message. This latter property is called *off-the-record* in the Designated Verifier Signature (DVS) literature [16, 24, 27–29, 31, 39–41], and is achieved by designing the scheme so that the receiver's secret key can be used to forge signatures. One may take this a step further and require *anonymity*, i.e. third parties cannot even learn who the sender and receiver are (this is called *privacy of identities* in the (M)DVS literature) [16].[3]

Another setting of interest is where the message is sent to many recipients. Consider, for example, the case of sending an email to multiple receivers. Apart from all the security properties listed above, here one would additionally require *consistency*: all the (intended) receivers will get the same email when decrypting the same ciphertext, even if the sender is dishonest. We note that it is crucial that a receiver can decrypt ciphertexts using only their secret key, i.e. without having to use the public key of the sender and other receivers. It is common in the literature to assume that the receiver knows who the sender and other receivers are so that their public keys can be used for decryption [6, 30]. But in many contexts adding this information in plain to the ciphertext would violate crucial properties, e.g., in broadcast encryption the ciphertext size would not be small any longer and in MDVS schemes anonymity (privacy) would be violated [30].

Many different schemes have been introduced in the literature that satisfy some of the properties listed here, see Sect. 1.5. In this work we propose two new primitives, Public Key Encryption for Broadcast (PKEBC) and Multi-Designated Receiver Signed Public Key Encryption (MDRS-PKE), which we explain in the following two subsections.

1.2 Public Key Encryption for Broadcast

The first type of primitive that we introduce, PKEBC, can be seen as an extension of Broadcast Encryption (BE) [19] which additionally gives consistency guarantees

[3] With off-the-record, a third party will know that either the alleged sender or the receiver wrote the message, whereas anonymity completely hides who the sender and receiver are. However, anonymity only holds when the receiver is honest whereas off-the-record provides guarantees against a dishonest receiver.

in the case of a dishonest sender.[4] More specifically, we expect PKEBC schemes to provide the following guarantees:

Correctness If a ciphertext c is honestly generated as the encryption of a message m with respect to a vector of receivers, say $\vec{R} \coloneqq (\text{Bob}, \text{Charlie})$, then we want that if Bob is honest and decrypts c using its secret key, it obtains a pair $((\mathtt{pk}_{\text{Bob}}, \mathtt{pk}_{\text{Charlie}}), m)$, where $\mathtt{pk}_{\text{Bob}}$ and $\mathtt{pk}_{\text{Charlie}}$ are, respectively, Bob's and Charlie's public keys;

Robustness Let c be the ciphertext from above. We do not want Dave, who is honest but yet not an intended receiver of c, to think c was meant for himself. In other words, we do not want Dave to successfully decrypt c.

Consistency Now consider a dishonest party Alice who wants to confuse Bob and Charlie, both of whom are honest. We do not want Alice to be capable of creating a ciphertext c such that when Bob decrypts c, it obtains some pair $((\mathtt{pk}_{\text{Bob}}, \mathtt{pk}_{\text{Charlie}}), m)$, but when Charlie decrypts c it obtains some different pair. Instead, we want that if Bob obtains a pair $((\mathtt{pk}_{\text{Bob}}, \mathtt{pk}_{\text{Charlie}}), m)$, then so will Charlie (and vice-versa).

Confidentiality Now, suppose that Alice is honest. If Alice encrypts a message m to Bob and Charlie (who are both honest), we do not want Eve, who is dishonest, to find out what m is.

Anonymity Finally, suppose there are two more honest receivers, say Frank and Grace, to whom Alice could also be sending a message to. If, again, Alice encrypts a message m to both Bob and Charlie, and letting c be the corresponding ciphertext, we do not want Eve to find out that the receivers of ciphertext c are Bob and Charlie; in fact, we do not want Eve to learn anything about the intended receivers of c, other than the number of receivers.

The formal definitions of PKEBC are given in Sect. 3. In Sect. 4 we show how to construct a PKEBC from standard assumptions. Our construction is a generalization of Naor-Yung's scheme [35] that enhances the security guarantees given by the original scheme. In particular, as we will see if the underlying PKE scheme is anonymous, then this anonymity is preserved by the PKEBC construction.

One important difference from other public key schemes for multiple parties is that to decrypt, a receiver only needs to know their own secret key;[5] the decryption of a ciphertext yields not only the underlying plaintext but also the set of receivers for the ciphertext. This then allows the corresponding public keys to be used as needed.[6]

[4] Though BE usually requires the ciphertext size to be sublinear in the number of receivers, which PKEBC does not.

[5] This problem has been noticed before. See [6, 30].

[6] We note that this is only important since we want to achieve anonymity, otherwise once could send the public keys of the other parties together with the ciphertext.

1.3 Multi-Designated Receiver Signed Public Key Encryption

Our main primitive has all of the properties listed in Sect. 1.1. Namely, a MDRS-PKE scheme is expected to provide the following guarantees:

Correctness If a ciphertext c is honestly generated as the encryption of a message m from a sender Alice to a vector of receivers $\vec{R} \coloneqq (\text{Bob}, \text{Charlie})$ then we want that if Bob is honest and decrypts c using its secret key, it obtains a triple $(\mathtt{spk}_{\text{Alice}}, (\mathtt{rpk}_{\text{Bob}}, \mathtt{rpk}_{\text{Charlie}}), m)$, where $\mathtt{spk}_{\text{Alice}}$ is Alice's public sending key, and $\mathtt{rpk}_{\text{Bob}}$ and $\mathtt{rpk}_{\text{Charlie}}$ are, respectively, Bob's and Charlie's receiver public keys;

Consistency Now consider a dishonest party Donald who is a sender and wants to confuse Bob and Charlie, both of whom are honest. We do not want Donald to be able to create a ciphertext c such that when Bob decrypts c, it obtains some triple $(\mathtt{spk}_{\text{Donald}}, (\mathtt{pk}_{\text{Bob}}, \mathtt{pk}_{\text{Charlie}}), m)$, but when Charlie decrypts c it obtains some different triple (or does not even decrypt). Instead, we want that if Bob obtains a triple $(\mathtt{spk}_{\text{Donald}}, (\mathtt{pk}_{\text{Bob}}, \mathtt{pk}_{\text{Charlie}}), m)$, then so will Charlie (and vice-versa).

Unforgeability We do not want that Eve can forge a ciphertext as if it were from an honest sender, say Alice, to a vector of receivers Bob and Charlie.

Confidentiality If an honest sender Alice encrypts a message m to Bob and Charlie (who are both honest), we do not want Eve, who is dishonest, to find out what m is.

Anonymity Suppose there is another honest sender, say Heidi. If Alice encrypts a message m to Bob, and letting c be the corresponding ciphertext, we do not want Eve to find out that Alice is the sender or that Bob is the receiver; Eve should at most learn that someone sent a message to a single receiver.

Off-The-Record Suppose Alice sends a message to Bob, Charlie and Donald. Donald, being dishonest, might be enticed to try convincing Eve that Alice sent some message. However, we do not want Donald to have this capability.

The formal definitions of MDRS-PKE are given in Sect. 5. In Sect. 6 we show how to construct a MDRS-PKE from standard assumptions. As we will see, our construction essentially consists of using the MDVS scheme to sign messages, using the PKEBC scheme to encrypt the signed messages, together with their MDVS signatures, and then using a Digital Signature Scheme (DSS) to bind the PKEBC ciphertext and the MDVS signature together.

Since an MDRS-PKE scheme is an extension of an MDVS scheme with privacy of identities and confidentiality, for completeness, we show in Appendix F that any MDRS-PKE scheme yields an MDVS scheme with privacy of identities. Since we give an MDRS-PKE scheme which is secure under standard assumptions, this in particular implies that our construction is the first achieving privacy of identities from standard assumptions. The only previous construction of an MDVS scheme with privacy of identities relied on a Verifiable Functional Encryption scheme for general circuits [16].

1.4 Applications to Secure (Group) Messaging

As we now discuss, one main application of MDRS-PKE schemes is secure messaging, and in particular secure group messaging.

Suppose Alice and Bob are using a secure messaging application to chat with each other. Of course, they expect the messenger to provide basic guarantees such as *Correctness*—if Alice sends a message to Bob, Bob receives this message—*Confidentiality*—no one other than Alice and Bob should learn the contents of the messages—and *Authenticity*—if Alice reads a message m, then Bob must have sent m. Another desirable guarantee they could expect from the messenger is *Anonymity*: suppose that in parallel to Alice and Bob's chat, Charlie and Dave are also chatting; then, if a third party Eve intercepts a ciphertext c from Alice and Bob's chat and Eve cannot *a priori* tell that c came from and/or is addressed to Alice or Bob, then Eve should not gain any additional information about the identity of c's sender and/or receiver from inspecting the contents of ciphertext c itself (in other words, Eve cannot tell if the ciphertext is from Alice and Bob's chat, from Alice and Charlie's chat, from Bob and Charlie's chat, or from Charlie and Dave's chat). Finally, imagine that Bob, who wants to keep the history of his chat with Alice, outsources the storage of the chat's ciphertexts to an external storage service which reliably, but not authentically, stores these ciphertexts. An important additional guarantee Alice expects from the messaging application is *Off-The-Record Deniability* (*Off-The-Record*) [11, 16]: if, somehow, Eve manages to access whatever is stored by Bob's storage service, Eve cannot tell by inspecting the stored ciphertexts, even if Bob chooses to cooperate with Eve[7], if these ciphertexts are authentic ones corresponding to real messages sent by Alice to Bob in their chat, or if they are fake ones generated by Bob (in case Bob is cooperating with Eve) or generated by anyone else (in case Bob is not cooperating with Eve) to incriminate Alice.

A related, yet very different property that secure messaging applications like Signal [15] provide is *Forward Secrecy* [25]. Informally, Forward Secrecy guarantees that even if Eve stores any ciphertexts received by Bob and later *hacks* into Bob's computer to learn his secret key, Eve cannot learn the decryptions (i.e. the plaintexts) of the ciphertexts she previously intercepted. Off-The-Record, on the other hand, does not give any guarantees about hiding the contents of previously exchanged messages. However, it hides from Eve whether Alice really sent a message m to Bob or if Bob faked receiving m. Furthermore, Forward Secrecy assumes Bob is honest: if Bob were dishonest, he could simply store the decryptions of the ciphertexts he receives to later disclose them to Eve. Off-The-Record does not make such assumption: even if Bob is dishonest, Eve cannot tell if it was Alice sending a message m, or if Bob faked receiving m from Alice (in case Bob is dishonest), or anyone else faked Alice sending m to Bob (in case Bob is honest). Finally, as one can deduce, Forward Secrecy is incompatible with parties keeping a history of their chats, whereas this is not the case for Off-The-Record. A different problem is Alice's computer getting *hacked* by Eve.

[7] By Bob collaborating with Eve we mean that Bob shares all his secrets (including secret keys) with Eve.

In such scenario it would be desirable to still give the Off-The-Record guarantee to Alice: Eve should not be able to tell if Alice ever sent any message or not. However, current Off-The-Record notions [16], including the one given in this paper, do not capture this.

A natural generalization of two party secure messaging is secure group messaging [2, 16]. Suppose Alice, Bob and Charlie now share a group chat. The key difference between Alice, Bob and Charlie sharing a group chat or having multiple two party chats with each other is *Consistency*: even if Charlie is dishonest, he cannot create confusion among Alice and Bob as to whether he sent a message to the group chat or not [16]. In other words, honest group members have a consistent view of the chat. Surprisingly, for the case of MDVS, this guarantee was only recently introduced by Damgård et al. in [16].

To achieve Off-The-Record in the group messaging case, one must consider that any subset of the parties participating in the group chat may be dishonest [16]. This property, also known as *Any-Subset Off-The-Record Deniability* (or more simply *Off-The-Record*) was first introduced by Damgård et al. in [16]. Returning to Alice, Bob and Charlie's group chat, this property essentially guarantees that regardless of who (among Bob and Charlie) cooperate with Eve in trying to convince her that Alice sent some message, Eve will not be convinced because any of them (or the two together) could have created a fake message to pretend that Alice sent it.

1.5 Related Work

A closely related type of encryption scheme are Broadcast Encryption (BE) schemes [10, 19]. However, BE schemes do not give the consistency guarantee that PKEBC give; the main goal of BE schemes is actually making ciphertexts short—ideally the size of ciphertexts would be independent from the number recipients. Conversely, the size of the ciphertexts of the PKEBC scheme construction we give in this paper grows quadratically with the number of recipients.

Another line of work, initiated by Diament et al., considered a special type of BE schemes, called Dual-Receiver Encryption (DRE) schemes, which allow a sender to send messages to two (and only two) receivers [17]. In subsequent work, Chow et al. introduce a notion of Soundness for DRE schemes [14]—which resembles our consistency notion when we restrict ourselves to the two party case—and show that the DRE scheme given in [17] satisfies their soundness notion. Chow et al. also introduce Dual-Receiver Key Encapsulation Mechanisms (DKEM)—a Key Encapsulation variant of DRE schemes that allows a sender to encapsulate a key for two receivers—along with a soundness notion that is analogous to the one for DRE schemes. In [21], Gegier shows how to construct DKEM schemes using Deterministic Encryption schemes. In contrast to PKEBC schemes, however, these schemes require the public keys of the intended receivers for decryption and do not give anonymity guarantees.

As already mentioned, PKEBC schemes allow receivers to decrypt a ciphertext meant for multiple receivers using their secret key only. This problem had been noticed before by Barth et al. in [6], and by Libert et al. in [30]. Barth et al.

modify the definition of BE schemes in a way that allows receivers to decrypt ciphertexts without knowing who the other recipients are a priori [6]. Libert et al. strengthens this by guaranteeing that receivers do not learn who the other receivers are, even after decrypting ciphertexts.

Other closely related works are Multi-Designated Verifier Signature (MDVS) schemes [16]. They provide consistency, authenticity, and off-the-record and sometimes also anonymity (called privacy). However, to the best of our knowledge, MDVS schemes require the public keys of the sender and other designated receivers to be used to verify signatures, and the existing literature does not discuss how the receiver gets that information, e.g. sending this information in plain would violate privacy. Thus, existing constructions of MDVS with privacy can only be used if the number of combinations of possible sender and receivers is small enough that all combinations can be tried by the verifier.

2 Preliminaries

We now introduce conventions and notation we use throughout the paper. We denote the arity of a vector $\vec{x}$ by $|\vec{x}|$ and its i-th element by x_i. We write $\alpha \in \vec{x}$ to denote $\exists i \in \{1, \ldots, |\vec{x}|\}$ with $\alpha = x_i$. We write $\mathrm{Set}(\vec{x})$ to denote the set induced by vector $\vec{x}$, i.e. $\mathrm{Set}(\vec{x}) \coloneqq \{x_i \mid x_i \in \vec{x}\}$.

Throughout the paper we frequently use vectors. We use upper case letters to denote vectors of parties, and lower case letters to denote vectors of artifacts such as public keys, messages, sequences of random coins, and so on. Moreover, we use the convention that if $\vec{V}$ is a vector of parties, then $\vec{v}$ denotes $\vec{V}$'s corresponding vector of public keys. For example, for a vector of parties $\vec{V} \coloneqq (\mathrm{Bob}, \mathrm{Charlie})$, $\vec{v} \coloneqq (\mathtt{pk}_{\mathrm{Bob}}, \mathtt{pk}_{\mathrm{Charlie}})$ is $\vec{V}$'s corresponding vector of public keys. In particular, V_1 is Bob and v_1 is Bob's public key $\mathtt{pk}_{\mathrm{Bob}}$, and V_2 is Charlie and v_2 is Charlie's public key $\mathtt{pk}_{\mathrm{Charlie}}$. More generally, for a vector of parties $\vec{V}$ with corresponding vector of public keys $\vec{v}$, V_i's public key is v_i, for $i \in \{1, \ldots, |\vec{V}|\}$.

3 Public Key Encryption for Broadcast Schemes

We now introduce the first new type of scheme we give in this paper, namely Public Key Encryption for Broadcast (PKEBC). A PKEBC scheme Π with message space $\mathcal{M}$ is a quadruple $\Pi = (S, G, E, D)$ of Probabilistic Polynomial Time Algorithms (PPTs), where:

- S: on input 1^k, generates public parameters $\mathtt{pp}$;
- G: on input $\mathtt{pp}$, generates a receiver key-pair;
- E: on input $(\mathtt{pp}, \vec{v}, m)$, where $\vec{v}$ is a vector of public keys of the intended receivers and m is the message, generates a ciphertext c;
- D: on input $(\mathtt{pp}, \mathtt{sk}, c)$, where $\mathtt{sk}$ is the receiver's secret key, D decrypts c using $\mathtt{sk}$, and outputs the decrypted receiver-vector/message pair $(\vec{v}, m)$ (or $\perp$ if the ciphertext did not decrypt correctly).

3.1 The Security of PKEBC Schemes

We now state the definitions of Correctness, Robustness, Consistency, and IND-CCA-2 and IK-CCA-2 security for PKEBC schemes. Before proceeding to the actual definitions, we first introduce some oracles the game systems from Definitions 1, 2 and 3 use. In the following, consider a PKEBC scheme $\Pi = (S, G, E, D)$ with message space $\mathcal{M}$. The oracles below are defined for a game-system with (an implicitly defined) security parameter k:

Public Parameters Oracle: $\mathcal{O}_{PP}$

1. On the first call, compute and store $\mathtt{pp} \leftarrow S(1^k)$; output $\mathtt{pp}$;
2. On subsequent calls, output the previously generated $\mathtt{pp}$.

Secret Key Generation Oracle: $\mathcal{O}_{SK}(B_j)$

1. If $\mathcal{O}_{SK}$ was queried on B_j before, simply look up and return the previously generated key for B_j;
2. Otherwise, store $(\mathtt{pk}_j, \mathtt{sk}_j) \leftarrow G(\mathtt{pp})$ as B_j's key-pair, and output $(\mathtt{pk}_j, \mathtt{sk}_j)$.

Public Key Generation Oracle: $\mathcal{O}_{PK}(B_j)$

1. $(\mathtt{pk}_j, \mathtt{sk}_j) \leftarrow \mathcal{O}_{SK}(B_j)$;
2. Output $\mathtt{pk}_j$.

Encryption Oracle: $\mathcal{O}_E(\vec{V}, m)$

1. $\vec{v} \leftarrow (\mathcal{O}_{PK}(V_1), \ldots, \mathcal{O}_{PK}(V_{|\vec{V}|}))$;
2. Create and output a fresh encryption $c \leftarrow E_{\mathtt{pp},\vec{v}}(m)$.

In addition to the oracles above, the game systems from Definitions 1 and 2 further provide adversaries with access to the following oracles:

Decryption Oracle: $\mathcal{O}_D(B_j, c)$

1. Query $\mathcal{O}_{SK}(B_j)$ to obtain the corresponding secret-key $\mathtt{sk}_j$;
2. Decrypt c using $\mathtt{sk}_j$, $(\vec{v}, m) \leftarrow D_{\mathtt{pp},\mathtt{sk}_j}(c)$, and then output the resulting receivers-message pair $(\vec{v}, m)$, or $\perp$ (if $(\vec{v}, m) = \perp$, i.e. the ciphertext is not valid with respect to B_j's secret key).

Definition 1 (Correctness). *Consider the following game played between between an adversary* $\mathbf{A}$ *and game system* $\mathbf{G}^{\mathsf{Corr}}$*:*

– $\mathbf{A}^{\mathcal{O}_{PP},\mathcal{O}_{PK},\mathcal{O}_{SK},\mathcal{O}_E,\mathcal{O}_D}$

$\mathbf{A}$ *wins the game if there are two queries* q_E *and* q_D *to* $\mathcal{O}_E$ *and* $\mathcal{O}_D$*, respectively, where* q_E *has input* $(\vec{V}, m)$ *and* q_D *has input* (B_j, c)*, satisfying* $B_j \in \vec{V}$*, the input* c *in* q_D *is the output of* q_E*, the output of* q_D *is either* $\perp$ *or* $(\vec{v}', m')$ *with* $(\vec{v}, m) \neq (\vec{v}', m')$*, and* $\mathbf{A}$ *did not query* $\mathcal{O}_{SK}$ *on input* B_j*.*

The advantage of $\mathbf{A}$ *in winning the Correctness game, denoted* $Adv^{\mathsf{Corr}}(\mathbf{A})$*, is the probability that* $\mathbf{A}$ *wins game* $\mathbf{G}^{\mathsf{Corr}}$ *as described above.*

The following notion captures the guarantee that if a ciphertext c is an honestly generated ciphertext for a vector of receivers $\vec{V}$ (for some message), then no honest receiver B who is not one of the intended receivers of c can successfully decrypt c (i.e. if $B \notin \vec{V}$ then the decryption of c with B's secret key outputs $\perp$). As one might note, this notion is a variant of the weak robustness notion introduced in [1], but adapted to PKEBC schemes.

Definition 2 (Robustness). *Consider the following game played between an adversary* $\mathbf{A}$ *and game system* $\mathbf{G}^{\mathsf{Rob}}$*:*

– $\mathbf{A}^{\mathcal{O}_{PP},\mathcal{O}_{PK},\mathcal{O}_{SK},\mathcal{O}_E,\mathcal{O}_D}$

$\mathbf{A}$ *wins the game if there are two queries* q_E *and* q_D *to* $\mathcal{O}_E$ *and* $\mathcal{O}_D$*, respectively, where* q_E *has input* $(\vec{V}, m)$ *and* q_D *has input* (B_j, c)*, satisfying* $B_j \notin \vec{V}$*, the input* c *in* q_D *is the output of* q_E*, the output of* q_D *is* $(\vec{v}', m')$ *with* $(\vec{v}', m') \neq \perp$*, and* $\mathbf{A}$ *did not query* $\mathcal{O}_{SK}$ *on input* B_j*.*

The advantage of $\mathbf{A}$ *in winning the Robustness game is the probability that* $\mathbf{A}$ *wins game* $\mathbf{G}^{\mathsf{Rob}}$ *as described above, and is denoted* $Adv^{\mathsf{Rob}}(\mathbf{A})$*.*

Remark 1. Correctness and robustness are properties only relevant to honest parties. It is common in the literature to either define such security notions without any adversary or to consider a stronger adversary that is unbounded or has access to the honest parties' secret keys. We choose the weaker definitions above for two main reasons: first, it has been proven that analogous correctness and robustness notions [1, 5] for PKE schemes—also defined with respect to computationally bounded adversaries who are not given access to the secret keys of honest parties—imply (corresponding) composable security notions (see [5] and [26]); second, since the remaining PKEBC security notions (e.g. IND-CCA-2 security) are defined with respect to computationally bounded adversaries that cannot obtain the secret keys of honest parties, there is no advantage in considering strengthened correctness and robustness security notions. Nevertheless, we note that our security proofs actually consider such stronger correctness and robustness notions (i.e. we prove our PKEBC construction satisfies such stronger notions).

We now introduce the notion of consistency. Essentially, this notion captures the guarantee that a dishonest sender cannot create confusion between any pair of honest receivers as to whether they received some message m with respect to a vector of receivers $\vec{V}$ that includes both parties.

Definition 3 (Consistency). *Consider the following game played between an adversary* $\mathbf{A}$ *and game system* $\mathbf{G}^{\mathsf{Cons}}$*:*

– $\mathbf{A}^{\mathcal{O}_{PP},\mathcal{O}_{PK},\mathcal{O}_{SK},\mathcal{O}_D}$

$\mathbf{A}$ *wins the game if there is a ciphertext* c *such that* $\mathcal{O}_D$ *is queried on inputs* (B_i, c) *and* (B_j, c) *for some* B_i *and* B_j *(possibly with* $B_i = B_j$*), there is no prior*

query on either B_i or B_j to $\mathcal{O}_{SK}$, query $\mathcal{O}_D(B_i, c)$ outputs some $(\vec{v}, m)$ satisfying $(\vec{v}, m) \neq \perp$ with $\mathtt{pk}_j \in \vec{v}$ (where $\mathtt{pk}_j$ is B_j's public key), and query $\mathcal{O}_D(B_j, c)$ does not output $(\vec{v}, m)$.

The advantage of $\mathbf{A}$ in winning the Consistency game is denoted $Adv^{\mathsf{Cons}}(\mathbf{A})$ and corresponds to the probability that $\mathbf{A}$ wins game $\mathbf{G}^{\mathsf{Cons}}$ as described above.

Remark 2. Similarly to Remark 1, Consistency is a security property only relevant to honest receivers, for which reason Definition 3 disallows adversaries from querying for the secret keys of honest receivers. It was proven in [32] that an analogous Consistency notion for MDVS schemes (introduced in [16]) implies composable security. Yet, and similarly to the case of Correctness and Robustness, our security proof actually shows something stronger: that our PKEBC construction is provably secure with respect to such a stronger consistency notion (in which the adversary can query for any party's secret key and still win the game).

The two following security notions are the multi-receiver variants of IND-CCA-2 security (introduced in [36]) and IK-CCA-2 security (introduced in [7]). The games defined by these notions provide adversaries with access to the oracles $\mathcal{O}_{PP}$ and $\mathcal{O}_{PK}$ defined above as well as to oracles $\mathcal{O}_E$ and $\mathcal{O}_D$. For both notions, $\mathcal{O}_D$ is defined as follows:

Decryption Oracle: $\mathcal{O}_D(B_j, c)$

1. If c was the output of some query to $\mathcal{O}_E$, output $\mathtt{test}$;
2. Otherwise, compute and output $(\vec{v}, m) \leftarrow D_{\mathtt{pp},\mathtt{sk}_j}(c)$, where $\mathtt{sk}_j$ is B_j's secret key.

The $\mathcal{O}_E$ oracle provided by the IND-CCA-2 games differs from the one provided by the IK-CCA-2 games; for IND-CCA-2, $\mathcal{O}_E$ is as follows:

Encryption Oracle: $\mathcal{O}_E(\vec{V}, m_0, m_1)$

1. For game system $\mathbf{G}_{\mathbf{b}}^{\mathsf{IND\text{-}CCA\text{-}2}}$, encrypt $m_{\mathbf{b}}$ under $\vec{v}$ (the vector of public keys corresponding to $\vec{V}$); output c.

Adversaries do not have access to $\mathcal{O}_{SK}$ in either notion.

Definition 4 (IND-CCA-2 Security). *Consider the following game played between an adversary $\mathbf{A}$ and a game system $\mathbf{G}_{\mathbf{b}}^{\mathsf{IND\text{-}CCA\text{-}2}}$, with $\mathbf{b} \in \{0, 1\}$:*

– $b' \leftarrow \mathbf{A}^{\mathcal{O}_{PP}, \mathcal{O}_{PK}, \mathcal{O}_E, \mathcal{O}_D}$

$\mathbf{A}$ wins the game if $b' = \mathbf{b}$ and every query $\mathcal{O}_E(\vec{V}, m_0, m_1)$ satisfies $|m_0| = |m_1|$.

We define the advantage of $\mathbf{A}$ in winning the IND-CCA-2 game as

$$Adv^{\mathsf{IND\text{-}CCA\text{-}2}}(\mathbf{A}) := \Big|\Pr[\mathbf{AG}_{\mathbf{0}}^{\mathsf{IND\text{-}CCA\text{-}2}} = \mathtt{win}] + \Pr[\mathbf{AG}_{\mathbf{1}}^{\mathsf{IND\text{-}CCA\text{-}2}} = \mathtt{win}] - 1\Big|.$$

For the IK-CCA-2 security notion, $\mathcal{O}_E$ behaves as follows:

Encryption Oracle: $\mathcal{O}_E(\vec{V}_0, \vec{V}_1, m)$

1. For game system $\mathbf{G}_{\mathbf{b}}^{\mathsf{IK\text{-}CCA\text{-}2}}$, encrypt m under $\vec{v}_{\mathbf{b}}$, the vector of public keys corresponding to $\vec{V}_{\mathbf{b}}$, creating a fresh ciphertext c; output c.

Definition 5 (IK-CCA-2 Security). *Consider the following game played between an adversary* $\mathbf{A}$ *and a game system* $\mathbf{G}_{\mathbf{b}}^{\mathsf{IK\text{-}CCA\text{-}2}}$*, with* $\mathbf{b} \in \{0,1\}$*:*

– $b' \leftarrow \mathbf{A}^{\mathcal{O}_{PP},\mathcal{O}_{PK},\mathcal{O}_E,\mathcal{O}_D}$

$\mathbf{A}$ *wins the game if* $b' = \mathbf{b}$ *and every query* $\mathcal{O}_E(\vec{V}_0, \vec{V}_1, m)$ *satisfies* $|\vec{V}_0| = |\vec{V}_1|$.
We define the advantage of $\mathbf{A}$ *in winning the* IK-CCA-2 *security game as*

$$Adv^{\mathsf{IK\text{-}CCA\text{-}2}}(\mathbf{A}) := \Big|\Pr[\mathbf{AG}_{\mathbf{0}}^{\mathsf{IK\text{-}CCA\text{-}2}} = \mathtt{win}] + \Pr[\mathbf{AG}_{\mathbf{1}}^{\mathsf{IK\text{-}CCA\text{-}2}} = \mathtt{win}] - 1\Big|.$$

We say that an adversary $\mathbf{A}$ (ε, t)-breaks the (n, d_E, q_E, q_D)-Correctness (resp. -Robustness, -Consistency, IND-CCA-2, IK-CCA-2) of Π if $\mathbf{A}$ runs in time at most t, queries the oracles it has access to on at most n different parties,[8] makes at most q_E and q_D queries to oracles $\mathcal{O}_E$ and $\mathcal{O}_D$, respectively, with the sum of lengths of all the party vectors input to $\mathcal{O}_E$ being at most d_E, and satisfies $Adv^{\mathsf{Corr}}(\mathbf{A}) \geq \varepsilon$ (resp. $Adv^{\mathsf{Rob}}(\mathbf{A}) \geq \varepsilon$, $Adv^{\mathsf{Cons}}(\mathbf{A}) \geq \varepsilon$, $Adv^{\mathsf{IND\text{-}CCA\text{-}2}}(\mathbf{A}) \geq \varepsilon$, $Adv^{\mathsf{IK\text{-}CCA\text{-}2}}(\mathbf{A}) \geq \varepsilon$). Finally, we say that Π is

$$(\varepsilon_{\mathsf{Corr}}, \varepsilon_{\mathsf{Rob}}, \varepsilon_{\mathsf{Cons}}, \varepsilon_{\mathsf{IND\text{-}CCA\text{-}2}}, \varepsilon_{\mathsf{IK\text{-}CCA\text{-}2}}, t, n, d_E, q_E, q_D)\text{-secure},$$

if no adversary $\mathbf{A}$:

– $(\varepsilon_{\mathsf{Corr}}, t)$-breaks the (n, d_E, q_E, q_D)-Correctness of Π;
– $(\varepsilon_{\mathsf{Rob}})$-breaks the (n, d_E, q_E, q_D)-Robustness of Π;
– $(\varepsilon_{\mathsf{Cons}}, t)$-breaks the (n, d_E, q_E, q_D)-Consistency of Π;
– $(\varepsilon_{\mathsf{IND\text{-}CCA\text{-}2}}, t)$-breaks the (n, d_E, q_E, q_D)-IND-CCA-2 security of Π; or
– $(\varepsilon_{\mathsf{IK\text{-}CCA\text{-}2}}, t)$-breaks the (n, d_E, q_E, q_D)-IK-CCA-2 security of Π.

4 A PKEBC Scheme from Standard Assumptions

We now present our construction of a PKEBC scheme. The construction is a generalization of Naor-Yung's scheme [35] that enhances the security guarantees given by the original scheme. In particular, if the underlying PKE scheme is anonymous, then this anonymity is preserved by the PKEBC construction. First, while the scheme should preserve the anonymity of the underlying PKE scheme, parties should still be able to obtain the vector of receivers from ciphertexts,

[8] Here, querying on most n parties means that the number of different parties in all queries is at most n. In particular, the number of different parties in a query $\mathcal{O}_E((B_1, B_2, B_3), (\ldots))$ is 3, assuming $B_1 \neq B_2 \neq B_3 \neq B_1$; the number of different parties in a query $\mathcal{O}_D(B_j, \cdot)$ is 1.

using only their own secret key. For this reason, the underlying PKE scheme is used to encrypt not only the messages to be sent, but also the vector of receivers to which each message is being sent to. As one might note, however, to preserve the anonymity of the underlying PKE scheme, the NIZK proof that proves the consistency of the ciphertexts for the various receivers can no longer be a proof for a statement in which the public keys are part of the statement. This introduces an extra complication since for some PKE schemes such as ElGamal, for every ciphertext c and message m, there is a public key $\mathtt{pk}$ and a sequence of random coins r such that c is an encryption of m under $\mathtt{pk}$, using r as the sequence of random coins for encrypting m. In particular, this means that the NIZK proof is not actually proving the consistency of the ciphertexts. To solve this issue, we further add a (binding) commitment to the vector of receiver public keys used to encrypt each ciphertext, and then use the NIZK proof to show that each ciphertext is an encryption of this same message under the public keys of the vector to which the commitment is bound. Note, however, that this is still not sufficient: despite now having the guarantee that if the NIZK proof verifies then all ciphertexts are encryptions of the same plaintext with respect a vector of public keys, since a party can still decrypt ciphertexts not meant for itself without realizing it, it could happen that a receiver decrypts the wrong ciphertext, thus getting the wrong vector of receivers-plaintext pair. To avoid this, the commitment additionally commits to the message being sent, and the sequence of random coins used to create the commitment are now encrypted along with the vector of public keys of the parties and the message being sent. This then allows a receiver to recompute the commitment from the vector of parties and message it decrypted. Given the commitment is binding, this implies that if the recomputed commitment matches the one in the ciphertext then decryption worked correctly (as otherwise the recomputed commitment would not match the one in the ciphertext).

We note that our security reductions are tight, and that there are tightly secure instantiations of each of the schemes we use as building blocks for our construction. For instance, ElGamal could be used as the underlying $\mathsf{IND\text{-}CPA}$ secure encryption scheme, as it is tightly multi-user multi-challenge $\mathsf{IND\text{-}CPA}$ secure [8]. For completeness, we show in the appendix that ElGamal is also tightly multi-user multi-challenge $\mathsf{IK\text{-}CPA}$ secure under the DDH assumption (see Appendix G). Furthermore, we could also use ElGamal as the statistically binding commitment scheme needed by our construction, and the tightly unbounded simulation sound NIZK scheme from [20].

Algorithm 1 gives a construction of a Public Key Encryption for Broadcast scheme $\Pi = (S, G, E, D)$ from a Public Key Encryption scheme $\Pi_{\mathrm{PKE}} = (G, E, D)$, a Commitment Scheme $\Pi_{\mathrm{CS}} = (G_{CRS}, \mathit{Commit}, \mathit{Verify})$ and a Non Interactive Zero Knowledge scheme $\Pi_{\mathrm{NIZK}} = (G_{CRS}, \mathit{Prove}, \mathit{Verify}, S \coloneqq (S_{CRS}, S_{Sim}))$. Con-

sider relation R_{Cons} defined as

$$\begin{aligned} R_{\text{Cons}} \coloneqq \Big\{ \big((\mathtt{crs}_{\text{CS}}, \mathtt{comm}, \vec{c}), (\rho, \vec{v}, m, \vec{r})\big) \mid \\ |\vec{c}| = |\vec{v}| \\ \wedge\, \mathtt{comm} = \Pi_{\text{CS}}.Commit_{\mathtt{crs}}(\vec{v}, m; \rho) \\ \wedge \big(\forall j \in \{1, \ldots, |\vec{c}|\}, \forall b \in \{0,1\}, \\ c_{j,b} = \Pi_{\text{PKE}}.E_{v_{j,b}}(\rho, \vec{v}, m; r_{j,b})\big)\Big\}. \end{aligned} \tag{4.1}$$

In Algorithm 1, we consider the language induced by R_{Cons}, which is defined as

$$\begin{aligned} L_{\text{Cons}} \coloneqq \{(\mathtt{crs}_{\text{CS}}, \mathtt{comm}, \vec{c}) \mid \\ \exists(\rho, \vec{v}, m, \vec{r}) \\ \big((\mathtt{crs}_{\text{CS}}, \mathtt{comm}, \vec{c}), (\rho, \vec{v}, m, \vec{r})\big) \in R_{\text{Cons}}\}. \end{aligned} \tag{4.2}$$

Algorithm 1 Construction of a PKEBC scheme $\Pi = (S, G, E, D)$.

$S(1^k)$
  return $(1^k, \Pi_{\text{NIZK}}.G_{CRS}(1^k), \Pi_{\text{CS}}.G_{CRS}(1^k))$

$G(\mathtt{pp} \coloneqq (1^k, \mathtt{crs}_{\text{NIZK}}, \mathtt{crs}_{\text{CS}}))$
  $(\mathtt{pk}_0, \mathtt{sk}_0) \leftarrow \Pi_{\text{PKE}}.G(1^k)$
  $(\mathtt{pk}_1, \mathtt{sk}_1) \leftarrow \Pi_{\text{PKE}}.G(1^k)$
  return $\big(\mathtt{pk} \coloneqq (\mathtt{pk}_0, \mathtt{pk}_1), \mathtt{sk} \coloneqq ((\mathtt{pk}_0, \mathtt{sk}_0), (\mathtt{pk}_1, \mathtt{sk}_1))\big)$

$E(\mathtt{pp} \coloneqq (1^k, \mathtt{crs}_{\text{NIZK}}, \mathtt{crs}_{\text{CS}}), \vec{v} \coloneqq \big((\mathtt{pk}_{1,0}, \mathtt{pk}_{1,1}), \ldots, (\mathtt{pk}_{|\vec{v}|,0}, \mathtt{pk}_{|\vec{v}|,1})\big), m \in \mathcal{M})$
  $\rho \leftarrow RandomCoins$
  $\mathtt{comm} \leftarrow \Pi_{\text{CS}}.Commit_{\mathtt{crs}_{\text{CS}}}(\vec{v}, m; \rho)$
  for $(\mathtt{pk}_{j,0}{}', \mathtt{pk}_{j,1}{}') \in \vec{v}$ **do**
    $(r_{j,0}, r_{j,1}) \leftarrow (RandomCoins, RandomCoins)$
    $(c_{j,0}, c_{j,1}) \leftarrow (\Pi_{\text{PKE}}.E_{\mathtt{pk}_{j,0}}(\rho, \vec{v}, m; r_{j,0}), \Pi_{\text{PKE}}.E_{\mathtt{pk}_{j,1}}(\rho, \vec{v}, m; r_{j,1}))$
  $\vec{r} \coloneqq \big((r_{1,0}, r_{1,1}), \ldots, (r_{|\vec{v}|,0}, r_{|\vec{v}|,1})\big)$
  $\vec{c} \coloneqq \big((c_{1,0}, c_{1,1}), \ldots, (c_{|\vec{v}|,0}, c_{|\vec{v}|,1})\big)$
  $p \leftarrow \Pi_{\text{NIZK}}.Prove_{\mathtt{crs}_{\text{NIZK}}}\big((\mathtt{crs}_{\text{CS}}, \mathtt{comm}, \vec{c}) \in L_{\text{Cons}}, (\vec{v}, m, \rho, \vec{r})\big)$
  return $(p, \mathtt{comm}, \vec{c})$

$D(\mathtt{pp} \coloneqq (1^k, \mathtt{crs}_{\text{NIZK}}, \mathtt{crs}_{\text{CS}}), \mathtt{sk}_j \coloneqq \big((\mathtt{pk}_{j,0}, \mathtt{sk}_{j,0}), (\mathtt{pk}_{j,1}, \mathtt{sk}_{j,1})\big), c \coloneqq (p, \mathtt{comm}, \vec{c}))$
  if $\Pi_{\text{NIZK}}.Verify_{\mathtt{crs}_{\text{NIZK}}}\big((\mathtt{crs}_{\text{CS}}, \mathtt{comm}, \vec{c}) \in L_{\text{Cons}}, p\big) = \mathtt{valid}$ **then**
    for $i \in \{1, \ldots, |\vec{c}|\}$ **do**
      $\big(\rho, \vec{v} \coloneqq \big((\mathtt{pk}_{1,0}{}', \mathtt{pk}_{1,1}{}'), \ldots, (\mathtt{pk}_{|\vec{v}|,0}{}', \mathtt{pk}_{|\vec{v}|,1}{}')\big), m\big) \leftarrow \Pi_{\text{PKE}}.D_{\mathtt{sk}_{j,0}{}'}(c_{i,0})$
      if $(\rho, \vec{v}, m) \neq \perp \wedge (\mathtt{pk}_{j,0}, \mathtt{pk}_{j,1}) = (\mathtt{pk}_{i,0}{}', \mathtt{pk}_{i,1}{}')$ **then**
        if $\mathtt{comm} = \Pi_{\text{CS}}.Commit_{\mathtt{crs}_{\text{CS}}}(\vec{v}, m; \rho)$ **then**
          return $(\vec{v}, m)$
  return $\perp$

4.1 Security Analysis of PKEBC Construction

Due to space constraints, the full proofs of the following results are in the appendix (see Appendix H).

Theorem 1. *If* Π_{PKE} *is*

$$(\varepsilon_{\text{PKE-IND-CPA}}, \varepsilon_{\text{PKE-IK-CPA}}, t_{\mathrm{PKE}}, n_{\mathrm{PKE}}, q_{E\,\mathrm{PKE}})\text{-}secure, \tag{4.3}$$

Π_{NIZK} *is*

$$\begin{aligned}(\varepsilon_{\text{NIZK-Complete}}, \varepsilon_{\text{NIZK-Sound}}, \varepsilon_{\text{NIZK-ZK}}, \varepsilon_{\text{NIZK-SS}},\\ t_{\mathrm{NIZK}}, q_{P\,\mathrm{NIZK}}, q_{V\,\mathrm{NIZK}})\text{-}secure,\end{aligned} \tag{4.4}$$

and Π_{CS} *is*

$$(\varepsilon_{\text{CS-Hiding}}, \varepsilon_{\text{CS-Binding}}, t_{\mathrm{CS}}, q_{\mathrm{CS}}, \mathsf{Binding})\text{-}secure, \tag{4.5}$$

then no adversary $\mathbf{A}$ (ε, t)*-breaks* Π*'s*

$$(q_E \coloneqq q_{P\,\mathrm{NIZK}}, q_D \coloneqq q_{V\,\mathrm{NIZK}})\text{-}Correctness,$$

with

$$\varepsilon > \varepsilon_{\text{CS-Binding}} + \varepsilon_{\text{NIZK-Complete}},$$

and $t_{\mathrm{NIZK}} \approx t + t_{\mathsf{Corr}}$*, where* t_{Corr} *is the time to run* Π*'s* $\mathbf{G}^{\mathsf{Corr}}$ *game.*

Remark 3. Theorem 1 states that Π's correctness holds against computationally bounded adversaries who do not have access to the secret keys of honest parties. However, since we use an underlying PKE with perfect correctness, the proof of Theorem 1 implies something stronger, namely that Π is correct according to a stronger correctness notion wherein adversaries are allowed to query for the secret key of any honest receiver.

Theorem 2. *If* Π_{PKE} *is*

$$(\varepsilon_{\text{PKE-IND-CPA}}, \varepsilon_{\text{PKE-IK-CPA}}, t_{\mathrm{PKE}}, n_{\mathrm{PKE}}, q_{E\,\mathrm{PKE}})\text{-}secure, \tag{4.6}$$

with $t_{\mathrm{PKE}} \gtrapprox n_{\mathrm{PKE}} \cdot t_G + t_D$ *(where* t_G *and* t_D *are, respectively, the times to run* $\Pi_{\mathrm{PKE}}.G$ *and* $\Pi_{\mathrm{PKE}}.D$*) and with* $n_{\mathrm{PKE}} \geq 1$*, and* Π_{CS} *is*

$$(\varepsilon_{\text{CS-Hiding}}, \varepsilon_{\text{CS-Binding}}, t_{\mathrm{CS}}, q_{\mathrm{CS}}, \mathsf{Binding})\text{-}secure, \tag{4.7}$$

then no adversary $\mathbf{A}$ (ε)*-breaks* Π*'s Robustness, with*

$$\varepsilon > 2 \cdot \varepsilon_{\text{PKE-IND-CPA}} + \varepsilon_{\text{CS-Binding}}.$$

Remark 4. Note that Theorem 2 states that Π's robustness holds against computationally unbounded adversaries; such adversaries can compute the private key of any party from its public key.

In the following we assume, without loss of generality for any practical purpose, that the NIZK proof verification algorithm is deterministic. For instance, the NIZK scheme given in [20] has deterministic proof verification and is tightly unbounded simulation sound. The reason for this assumption is that an adversary could potentially come up with a NIZK proof for a valid statement which would only be considered as valid by the NIZK verification algorithm sometimes.

Theorem 3. *If* Π_{PKE} *is*

$$(\varepsilon_{\text{PKE-IND-CPA}}, \varepsilon_{\text{PKE-IK-CPA}}, t_{\mathrm{PKE}}, n_{\mathrm{PKE}}, q_{E\,\mathrm{PKE}})\text{-}secure, \tag{4.8}$$

Π_{NIZK} *is*

$$\begin{aligned}(\varepsilon_{\text{NIZK-Complete}}, \varepsilon_{\text{NIZK-Sound}}, \varepsilon_{\text{NIZK-ZK}}, \varepsilon_{\text{NIZK-SS}},\\ t_{\mathrm{NIZK}}, q_{P\,\mathrm{NIZK}}, q_{V\,\mathrm{NIZK}})\text{-}secure,\end{aligned} \tag{4.9}$$

Π_{CS} *is*

$$(\varepsilon_{\text{CS-Hiding}}, \varepsilon_{\text{CS-Binding}}, t_{\mathrm{CS}}, q_{\mathrm{CS}}, \mathsf{Binding})\text{-}secure, \tag{4.10}$$

and $\Pi_{\mathrm{NIZK}}.V$ *is a deterministic algorithm, then no adversary* $\mathbf{A}$ (ε, t)*-breaks* Π*'s*

$$(q_D \coloneqq q_{V\,\mathrm{NIZK}})\text{-}Consistency,$$

with $\varepsilon > \varepsilon_{\text{CS-Binding}} + \varepsilon_{\text{NIZK-Sound}}$ *and with* $t_{\mathrm{NIZK}} \approx t + t_{\mathsf{Cons}}$*, where* t_{Cons} *is the time to run* Π*'s* $\mathbf{G}^{\mathsf{Cons}}$ *game.*

Remark 5. Theorem 3 states that Π's consistency holds against computationally bounded adversaries who do not have access to the secret keys of honest parties. However, similarly to Remark 3, its proof implies Π is consistent with respect to a stronger notion which allows adversaries to query for the secret key of any honest receiver.

Theorem 4. *If* Π_{PKE} *is*

$$(\varepsilon_{\text{PKE-IND-CPA}}, \varepsilon_{\text{PKE-IK-CPA}}, t_{\mathrm{PKE}}, n_{\mathrm{PKE}}, q_{E\,\mathrm{PKE}})\text{-}secure, \tag{4.11}$$

Π_{NIZK} *is*

$$\begin{aligned}(\varepsilon_{\text{NIZK-Complete}}, \varepsilon_{\text{NIZK-Sound}}, \varepsilon_{\text{NIZK-ZK}}, \varepsilon_{\text{NIZK-SS}},\\ t_{\mathrm{NIZK}}, q_{P\,\mathrm{NIZK}}, q_{V\,\mathrm{NIZK}})\text{-}secure,\end{aligned} \tag{4.12}$$

and Π_{CS} *is*

$$(\varepsilon_{\text{CS-Hiding}}, \varepsilon_{\text{CS-Binding}}, t_{\mathrm{CS}}, q_{\mathrm{CS}}, \mathsf{Binding})\text{-}secure, \tag{4.13}$$

then no adversary $\mathbf{A}$ (ε, t)*-breaks* Π*'s*

$$\begin{aligned}&(n \coloneqq n_{\mathrm{PKE}}, d_E \coloneqq q_{E\,\mathrm{PKE}},\\ &q_E \coloneqq \min(q_{P\,\mathrm{NIZK}}, q_{\mathrm{CS}}), q_D \coloneqq q_{V\,\mathrm{NIZK}})\text{-}\mathsf{IK\text{-}CCA\text{-}2}\ security,\end{aligned}$$

with

$$\begin{aligned}\varepsilon > 4 \cdot \varepsilon_{\text{PKE-IND-CPA}}&\\ + 2 \cdot (\varepsilon_{\text{NIZK-ZK}} + \varepsilon_{\text{PKE-IK-CPA}} + \varepsilon_{\text{NIZK-SS}})&\\ + \varepsilon_{\text{CS-Hiding}},&\\ t_{\mathrm{PKE}}, t_{\mathrm{CS}} \approx t + t_{\mathsf{IK\text{-}CCA\text{-}2}} + q_E \cdot t_{S_{Sim}} + t_{S_{CRS}},&\\ t_{\mathrm{NIZK}} \approx t + t_{\mathsf{IK\text{-}CCA\text{-}2}},&\end{aligned}$$

where $t_{\mathsf{IK\text{-}CCA\text{-}2}}$ *is the time to run* Π*'s* $\mathbf{G}_{\mathbf{b}}^{\mathsf{IK\text{-}CCA\text{-}2}}$ *game experiment,* $t_{S_{Sim}}$ *is the runtime of* S_{Sim}*, and* $t_{S_{CRS}}$ *is the runtime of* S_{CRS}*.*

Theorem 5. *If* Π_{PKE} *is*

$$(\varepsilon_{\text{PKE-IND-CPA}}, \varepsilon_{\text{PKE-IK-CPA}}, t_{\mathrm{PKE}}, n_{\mathrm{PKE}}, q_{E\,\mathrm{PKE}})\text{-}secure, \tag{4.14}$$

Π_{NIZK} *is*

$$\begin{aligned}(\varepsilon_{\text{NIZK-Complete}}, \varepsilon_{\text{NIZK-Sound}}, \varepsilon_{\text{NIZK-ZK}}, \varepsilon_{\text{NIZK-SS}},\\ t_{\mathrm{NIZK}}, q_{P\,\mathrm{NIZK}}, q_{V\,\mathrm{NIZK}})\text{-}secure,\end{aligned} \tag{4.15}$$

and Π_{CS} *is*

$$(\varepsilon_{\text{CS-Hiding}}, \varepsilon_{\text{CS-Binding}}, t_{\mathrm{CS}}, q_{\mathrm{CS}}, \mathsf{Binding})\text{-}secure, \tag{4.16}$$

then no adversary $\mathbf{A}$ (ε, t)*-breaks* Π*'s*

$$\begin{aligned}&(n \coloneqq n_{\mathrm{PKE}}, d_E \coloneqq q_{E\,\mathrm{PKE}},\\ &q_E \coloneqq \min(q_{P\,\mathrm{NIZK}}, q_{\mathrm{CS}}), q_D \coloneqq q_{V\,\mathrm{NIZK}})\text{-}\mathsf{IND\text{-}CCA\text{-}2}\ security,\end{aligned}$$

with

$$\begin{aligned}\varepsilon &> 4 \cdot \varepsilon_{\text{PKE-IND-CPA}}\\ &\quad + 2 \cdot (\varepsilon_{\text{NIZK-ZK}} + \varepsilon_{\text{NIZK-SS}})\\ &\quad + \varepsilon_{\text{CS-Hiding}}\\ t_{\mathrm{PKE}} &\approx t + t_{\mathsf{IND\text{-}CCA\text{-}2}} + q_E \cdot t_{S_{Sim}} + t_{S_{CRS}},\\ t_{\mathrm{NIZK}}, t_{\mathrm{CS}} &\approx t + t_{\mathsf{IND\text{-}CCA\text{-}2}},\end{aligned}$$

where $t_{\mathsf{IND\text{-}CCA\text{-}2}}$ *is the time to run* Π*'s* $\mathbf{G}_{\mathbf{b}}^{\mathsf{IND\text{-}CCA\text{-}2}}$ *game,* $t_{S_{Sim}}$ *is the runtime of* S_{Sim}*, and* $t_{S_{CRS}}$ *is the runtime of* S_{CRS}*.*

5 Multi-Designated Receiver Signed Public Key Encryption Schemes

We now introduce Multi-Designated Receiver Signed Public Key Encryption (MDRS-PKE) schemes. An MDRS-PKE scheme $\Pi = (S, G_S, G_R, E, D, Forge)$ with message space $\mathcal{M}$ is a six-tuple of PPTs, where:

- S: on input 1^k, generates public parameters $\mathtt{pp}$;
- G_S: on input $\mathtt{pp}$, generates a sender key-pair;
- G_R: on input $\mathtt{pp}$, generates a receiver key-pair;
- E: on input $(\mathtt{pp}, \mathtt{ssk}, \vec{v}, m)$, where $\mathtt{ssk}$ is the secret sending key, $\vec{v}$ is a vector of public keys of the intended receivers, and m is the message, generates a ciphertext c;
- D: on input $(\mathtt{pp}, \mathtt{rsk}, c)$, where $\mathtt{rsk}$ is the receiver's secret key, D decrypts c using $\mathtt{rsk}$, obtaining a triple sender/receiver-vector/message $(\mathtt{spk}, \vec{v}, m)$ (or $\perp$ if decryption fails) which it then outputs;

– *Forge*: on input $(\mathtt{pp}, \mathtt{spk}, \vec{v}, m, \vec{s})$, where $\mathtt{spk}$ is the sender's public key, $\vec{v}$ is a vector of public keys of the intended receivers, m is the message and $\vec{s}$ is a vector of designated receivers' secret keys—with $|\vec{s}| = |\vec{v}|$ and where for $i \in \{1, \ldots, |\vec{v}|\}$, either $s_i = \perp$ or s_i is the secret key corresponding to the i-th public key of $\vec{v}$, i.e. v_i—generates a ciphertext c.

5.1 The Security of MDRS-PKE Schemes

Below we state the definitions of Correctness, Consistency, Unforgeability, IND-CCA-2 security, IK-CCA-2 security and Off-The-Record for MDRS-PKE schemes. Before proceeding to the actual definitions, we first introduce some oracles the game systems for MDRS-PKE use. In the following, consider an MDRS-PKE scheme $\Pi = (S, G_S, G_R, E, D, \mathit{Forge})$ with message space $\mathcal{M}$. The oracles below are defined for a game-system with (an implicitly defined) security parameter k:

Public Parameter Generation Oracle: $\mathcal{O}_{PP}$

1. On the first call, compute $\mathtt{pp} \leftarrow S(1^k)$; output $\mathtt{pp}$;
2. On subsequent calls, simply output $\mathtt{pp}$.

Sender Key-Pair Oracle: $\mathcal{O}_{SK}(A_i)$

1. On the first call on input A_i, compute and store $(\mathtt{spk}_i, \mathtt{ssk}_i) \leftarrow G_S(\mathtt{pp})$; output $(\mathtt{spk}_i, \mathtt{ssk}_i)$;
2. On subsequent calls, simply output $(\mathtt{spk}_i, \mathtt{ssk}_i)$.

Receiver Key-Pair Oracle: $\mathcal{O}_{RK}(B_j)$

1. Analogous to the Sender Key-Pair Oracle.

Sender Public-Key Oracle: $\mathcal{O}_{SPK}(A_i)$

1. $(\mathtt{spk}_i, \mathtt{ssk}_i) \leftarrow \mathcal{O}_{SK}(A_i)$; output $\mathtt{spk}_i$.

Receiver Public-Key Oracle: $\mathcal{O}_{RPK}(B_j)$

1. Analogous to the Sender Public-Key Oracle.

Encryption Oracle: $\mathcal{O}_E(A_i, \vec{V}, m)$

1. $(\mathtt{spk}_i, \mathtt{ssk}_i) \leftarrow \mathcal{O}_{SK}(A_i)$;
2. $\vec{v} \leftarrow (\mathcal{O}_{RPK}(V_1), \ldots, \mathcal{O}_{RPK}(V_{|\vec{V}|}))$;
3. Output $c \leftarrow E_{\mathtt{pp}}(\mathtt{ssk}_i, \vec{v}, m)$.

Decryption Oracle: $\mathcal{O}_D(B_j, c)$

1. $(\mathtt{rpk}_j, \mathtt{rsk}_j) \leftarrow \mathcal{O}_{RK}(B_j)$;
2. $(\mathtt{spk}_i, \vec{v} \coloneqq (\mathtt{rpk}_1, \ldots, \mathtt{rpk}_{|\vec{v}|}), m) \leftarrow D_{\mathtt{pp}}(\mathtt{rsk}_j, c)$;

3. If, for each party A_i previously input to either $\mathcal{O}_{SK}$, $\mathcal{O}_{SPK}$ or $\mathcal{O}_E$, $\mathtt{spk}_i \neq \mathcal{O}_{SPK}(A_i)$, then output $\perp$;
4. If, for some $l \in \{1, \ldots, |\vec{V}|\}$, there is no party B_j that was previously input to either $\mathcal{O}_{RK}$, $\mathcal{O}_{RPK}$, $\mathcal{O}_E$ or $\mathcal{O}_D$ such that $v_l = \mathcal{O}_{RPK}(V_l)$, then output $\perp$;
5. Output $(\mathtt{spk}, \vec{v}, m)$.

We now introduce the game-based notions. Let $\Pi = (S, G_S, G_R, E, D, Forge)$ be an MDRS-PKE.

Definition 6 (Correctness). *Consider the following game played between an adversary* $\mathbf{A}$ *and game system* $\mathbf{G}^{\mathsf{Corr}}$*:*

– $\mathbf{A}^{\mathcal{O}_{PP}, \mathcal{O}_{SPK}, \mathcal{O}_{SK}, \mathcal{O}_{RPK}, \mathcal{O}_{RK}, \mathcal{O}_E, \mathcal{O}_D}$

$\mathbf{A}$ *wins the game if there is a query* q_E *to* $\mathcal{O}_E$ *and a later query* q_D *to* $\mathcal{O}_D$ *such that* q_E *has input* $(A_i, \vec{V}, m)$ *and* q_D *has input* (B_j, c) *with* $B_j \in \vec{V}$ *and* c *being the output of* q_E*, the output of* q_D *is* $(\mathtt{spk}_i{}', \vec{v}', m')$ *with* $(\mathtt{spk}_i{}', \vec{v}', m') \neq (\mathtt{spk}_i, \vec{v}, m)$*—where* $\mathtt{spk}_i$ *is* A_i*'s public key and* $\vec{v}$ *is the vector of public keys corresponding to* $\vec{V}$*—and* $\mathbf{A}$ *did not query* $\mathcal{O}_{SK}$ *on* A_i *nor* $\mathcal{O}_{RK}$ *on* B_j*.*

The advantage of $\mathbf{A}$ *in winning the Correctness game, denoted* $Adv^{\mathsf{Corr}}(\mathbf{A})$*, is the probability that* $\mathbf{A}$ *wins game* $\mathbf{G}^{\mathsf{Corr}}$ *as described above.*

As already noted in Remark 1, Correctness is a property only relevant to honest parties. As these parties are not corrupted, their keys do not leak to the adversary. Definition 6 hence disallows adversaries from querying for the secret keys of honest parties. Note that the analogous Correctness notion for MDVS schemes introduced in [32]—which also does not allow adversaries to query for the secret keys of honest parties—is known to imply the composable security of MDVS schemes (see [32]). As noted in Remark 10, the MDRS-PKE construction we give actually satisfies a stronger Correctness notion analogous to the one mentioned in Remark 1, as long as both of the underlying (PKEBC and MDVS) schemes satisfy analogous Correctness notions.

The following notion captures Consistency for MDRS-PKE schemes, and is analogous to the PKEBC Consistency notion.

Definition 7 (Consistency). *Consider the following game played between an adversary* $\mathbf{A}$ *and game system* $\mathbf{G}^{\mathsf{Cons}}$*:*

– $\mathbf{A}^{\mathcal{O}_{PP}, \mathcal{O}_{SPK}, \mathcal{O}_{SK}, \mathcal{O}_{RPK}, \mathcal{O}_{RK}, \mathcal{O}_E, \mathcal{O}_D}$

$\mathbf{A}$ *wins the game if there are two* $\mathcal{O}_D$ *queries, say* q_{D_i} *and* q_{D_j}*, on inputs, respectively,* (B_i, c) *and* (B_j, c') *with* $c = c'$ *such that:*

1. *the outputs of* q_{D_i} *and* q_{D_j} *differ;*
2. *either the receiver public key* $\mathtt{rpk}_j$ *of* B_j *is part of the vector of receiver public keys output by* q_{D_i}*, or the receiver public key* $\mathtt{rpk}_i$ *of* B_i *is part of the vector of public keys output by* q_{D_j}*;*

3. *there is no query* $\mathcal{O}_{RK}(B_i)$ *(resp.* $\mathcal{O}_{RK}(B_j)$*) prior to* q_{D_i} *(resp.* q_{D_j}*); and*
4. *there is no sender* A *(resp. no receiver* B*) which had not been input to a query* $\mathcal{O}_{SPK}$*,* $\mathcal{O}_{SK}$ *or* $\mathcal{O}_E$ *(resp.* $\mathcal{O}_{RPK}$*,* $\mathcal{O}_{RK}$ *or* $\mathcal{O}_E$*) prior to both* q_{D_i} *and* q_{D_j} *and whose public key is output by one of these queries.*

The advantage of $\mathbf{A}$ *in winning the Consistency game is denoted* $Adv^{\mathsf{Cons}}(\mathbf{A})$ *and corresponds to the probability that* $\mathbf{A}$ *wins game* $\mathbf{G}^{\mathsf{Cons}}$ *as described above.*

Remark 6. Note that $\mathcal{O}_D$ is defined in such a way that, if the MDRS-PKE decryption of some ciphertext outputs a sender public key that matches no previously queried sender's public key, or if one of the receiver public keys in the vector matches no previously queried receiver's public key, then the oracle outputs $\perp$. The last condition in the definition above ensures that the outputs of the two decryption queries are not different just because, by chance, there is a party—whose public key is in the MDRS-PKE decryption of the ciphertext—that had not been input to any query before the two queries q_i and q_j, but that was given as input to some query made between the q_i and q_j queries.[9]

The following security notion is analogous to the Unforgeability security notion for MDVS schemes. For the case of a single receiver, it informally states that if a sender A is honest, then no dishonest party can forge a ciphertext that fools an honest receiver into believing A sent some message that A actually did not send.

Definition 8 (Unforgeability). *Consider the following game played between adversary* $\mathbf{A}$ *and game system* $\mathbf{G}^{\mathsf{Unforg}}$*:*

– $\mathbf{A}^{\mathcal{O}_{PP},\mathcal{O}_{SPK},\mathcal{O}_{SK},\mathcal{O}_{RPK},\mathcal{O}_{RK},\mathcal{O}_E,\mathcal{O}_D}$

$\mathbf{A}$ *wins the game if it makes a query* $\mathcal{O}_D(B_j, c)$ *that outputs* $(\mathtt{spk}_i, \vec{v}, m) \neq \perp$*, there is a sender* A_i *and a vector of receivers* $\vec{V}$ *with* $B_j \in \vec{V}$ *such that* $\mathtt{spk}_i$ *is* A_i*'s sender public key (i.e.* $\mathcal{O}_{SPK}(A_i) = \mathtt{spk}_i$*) and* $\vec{v}$ *is the vector of receiver public keys corresponding to* $\vec{V}$ *(i.e.* $|\vec{V}| = |\vec{v}|$ *and for each* $l \in \{1, \ldots, |\vec{v}|\}$*,* $\mathcal{O}_{RPK}(V_l) = v_l$*), there was no query* $\mathcal{O}_E({A_i}', \vec{V}', m')$ *with* $({A_i}', \vec{V}', m') = (A_i, \vec{V}, m)$*, no query* $\mathcal{O}_{SK}(A_i)$ *and no query* $\mathcal{O}_{RK}(B_j)$*.*

The advantage of $\mathbf{A}$ *in winning the Unforgeability game is the probability that* $\mathbf{A}$ *wins game* $\mathbf{G}^{\mathsf{Unforg}}$ *as described above, and is denoted* $Adv^{\mathsf{Unforg}}(\mathbf{A})$*.*

The following security notions are the MDRS-PKE variants of Definitions 4 and 5. The games defined by these notions provide adversaries with access to the oracles $\mathcal{O}_{PP}$, $\mathcal{O}_{SPK}$, $\mathcal{O}_{SK}$ and $\mathcal{O}_{RPK}$ defined above as well as to oracles $\mathcal{O}_E$ and $\mathcal{O}_D$. For both notions, $\mathcal{O}_D$ is defined as follows:

[9] Looking ahead, this condition seems necessary to allow us to prove the Consistency of our MDRS-PKE construction (see 2): without this condition, we do not know how to reduce an adversary from breaking the Consistency of our MDRS-PKE construction to one breaking the consistency of the underlying MDVS (and another breaking the consistency of the underlying PKEBC scheme) because the verification oracle of the MDVS security notions only allows the adversary to input parties, not public keys.

Decryption Oracle: $\mathcal{O}_D(B_j, c)$

1. If c was the output of some query to $\mathcal{O}_E$, output `test`;
2. Otherwise, proceed as in the default $\mathcal{O}_D$ oracle.

The $\mathcal{O}_E$ oracle provided by the IND-CCA-2 games differs from the one provided by the IK-CCA-2 games; for IND-CCA-2, $\mathcal{O}_E$ is as follows:

Encryption Oracle: $\mathcal{O}_E(A_i, \vec{V}, m_0, m_1)$

1. For game system $\mathbf{G}_{\mathbf{b}}^{\mathsf{IND\text{-}CCA\text{-}2}}$, encrypt $m_{\mathbf{b}}$ under $\mathtt{ssk}_i$ (A_i's sender secret key) and $\vec{v}$ ($\vec{V}$'s corresponding vector of receiver public keys); output c.

Definition 9 (IND-CCA-2). *Consider the following game played between an adversary* $\mathbf{A}$ *and a game system* $\mathbf{G}_{\mathbf{b}}^{\mathsf{IND\text{-}CCA\text{-}2}}$*, with* $\mathbf{b} \in \{0, 1\}$*:*

– $b' \leftarrow \mathbf{A}^{\mathcal{O}_{PP}, \mathcal{O}_{SPK}, \mathcal{O}_{SK}, \mathcal{O}_{RPK}, \mathcal{O}_E, \mathcal{O}_D}$

$\mathbf{A}$ *wins the game if* $b' = \mathbf{b}$ *and for every query* $\mathcal{O}_E(A_i, \vec{V}, m_0, m_1)$*:*

– $|m_0| = |m_1|$*; and*
– *there is no query on* A_i *to* $\mathcal{O}_{SK}$*.*

We define the advantage of $\mathbf{A}$ *in winning the* IND-CCA-2 *game as*

$$Adv^{\mathsf{IND\text{-}CCA\text{-}2}}(\mathbf{A}) := \Big|\Pr[\mathbf{AG}_{\mathbf{0}}^{\mathsf{IND\text{-}CCA\text{-}2}} = \mathtt{win}] + \Pr[\mathbf{AG}_{\mathbf{1}}^{\mathsf{IND\text{-}CCA\text{-}2}} = \mathtt{win}] - 1\Big|.$$

For the IK-CCA-2 security notion, $\mathcal{O}_E$ behaves as follows:

Encryption Oracle: $\mathcal{O}_E((A_{i,0}, \vec{V}_0), (A_{i,1}, \vec{V}_1), m)$

1. For game system $\mathbf{G}_{\mathbf{b}}^{\mathsf{IK\text{-}CCA\text{-}2}}$, encrypt m under $\mathtt{ssk}_{i,\mathbf{b}}$ ($A_{i,\mathbf{b}}$'s secret key) and $\vec{v}_{\mathbf{b}}$ (the vector of public keys corresponding to $\vec{V}_{\mathbf{b}}$), creating a fresh ciphertext c; output c.

Definition 10 (IK-CCA-2). *Consider the following game played between an adversary* $\mathbf{A}$ *and a game system* $\mathbf{G}_{\mathbf{b}}^{\mathsf{IK\text{-}CCA\text{-}2}}$*, with* $\mathbf{b} \in \{0, 1\}$*:*

– $b' \leftarrow \mathbf{A}^{\mathcal{O}_{PP}, \mathcal{O}_{SPK}, \mathcal{O}_{SK}, \mathcal{O}_{RPK}, \mathcal{O}_E, \mathcal{O}_D}$

$\mathbf{A}$ *wins the game if* $b' = \mathbf{b}$ *and for every query* $((A_{i,0}, \vec{V}_0), (A_{i,1}, \vec{V}_1), m)$ *to* $\mathcal{O}_E$*:*

– $|\vec{V}_0| = |\vec{V}_1|$*; and*
– $\mathcal{O}_{SK}$ *is not queried on neither* $A_{i,0}$ *and* $A_{i,1}$*.*

We define the advantage of $\mathbf{A}$ *in winning the* IK-CCA-2 *security game as*

$$Adv^{\mathsf{IK\text{-}CCA\text{-}2}}(\mathbf{A}) := \Big|\Pr[\mathbf{AG}_{\mathbf{0}}^{\mathsf{IK\text{-}CCA\text{-}2}} = \mathtt{win}] + \Pr[\mathbf{AG}_{\mathbf{1}}^{\mathsf{IK\text{-}CCA\text{-}2}} = \mathtt{win}] - 1\Big|.$$

Remark 7. The IND-CCA-2 and IK-CCA-2 notions for MDRS-PKE schemes capture, respectively, confidentiality and anonymity. Even though one could define stronger variants of these notions wherein the adversary is allowed to query for the secret key of any sender, we chose these definitions because they are weaker, but yet strong enough to imply composable security (see [3, 4, 22] for the analogous case of the Outsider Security Model for Signcryption). Nonetheless, our MDRS-PKE construction satisfies the stronger IND-CCA-2 and IK-CCA-2 security notions in which the adversary is allowed to query for the secret key of every sender.[10]

The following notion captures the Off-The-Record property of MDRS-PKE schemes, and resembles the (Any-Subset) Off-The-Record security notion introduced in [16] for MDVS schemes. This notion defines two game systems, $\mathbf{G}_{\mathbf{0}}^{\mathsf{OTR}}$ and $\mathbf{G}_{\mathbf{1}}^{\mathsf{OTR}}$. The game systems also provide adversaries with access to oracles $\mathcal{O}_E$ and $\mathcal{O}_D$ defined below:

Encryption Oracle: $\mathcal{O}_E(\mathtt{type} \in \{\mathtt{sig}, \mathtt{sim}\}, A_i, \vec{V}, m, \mathcal{C} \subseteq \mathrm{Set}(\vec{V}))$

For $\mathbf{b} \in \{0, 1\}$, oracle $\mathcal{O}_E$ of game system $\mathbf{G}_{\mathbf{b}}^{\mathsf{OTR}}$ behaves as follows:

1. Let $(\mathtt{spk}_i, \mathtt{ssk}_i) \leftarrow \mathcal{O}_{SK}(A_i)$;
2. Let $\vec{v} = (v_1, \ldots, v_{|\vec{V}|})$ and $\vec{s} = (s_1, \ldots, s_{|\vec{V}|})$, where, for $i = 1, \ldots, |\vec{V}|$:
   - $(v_i, s_i) = \begin{cases} \mathcal{O}_{RK}(V_i) & \text{if } V_i \in \mathcal{C} \\ (\mathcal{O}_{RPK}(V_i), \perp) & \text{otherwise;} \end{cases}$
3. $(c_{\mathbf{0}}, c_{\mathbf{1}}) \leftarrow (\Pi.E_{\mathtt{pp}}(\mathtt{ssk}_i, \vec{v}, m), \Pi.Forge_{\mathtt{pp}}(\mathtt{spk}_i, \vec{v}, m, \vec{s}))$;
4. If $\mathbf{b} = 0$, output $c_{\mathbf{0}}$ if $\mathtt{type} = \mathtt{sig}$ and $c_{\mathbf{1}}$ if $\mathtt{type} = \mathtt{sim}$;
5. Otherwise, if $\mathbf{b} = 1$, output $c_{\mathbf{1}}$.

Decryption Oracle: $\mathcal{O}_D(B_j, c)$

1. If c was the output of some query to $\mathcal{O}_E$, output $\mathtt{test}$;
2. Otherwise, proceed as in the default $\mathcal{O}_D$ oracle.

Definition 11 (Off-The-Record). *Consider the following game played between an adversary* $\mathbf{A}$ *and game system* $\mathbf{G}_{\mathbf{b}}^{\mathsf{OTR}}$*:*

- $b' \leftarrow \mathbf{A}^{\mathcal{O}_{PP}, \mathcal{O}_{SPK}, \mathcal{O}_{SK}, \mathcal{O}_{RPK}, \mathcal{O}_{RK}, \mathcal{O}_E, \mathcal{O}_D}$

$\mathbf{A}$ *wins the game if it outputs a guess bit* b' *with* $b' = \mathbf{b}$ *and for every query* $\mathcal{O}_E(\mathtt{type}, A_i, \vec{V}, m, \mathcal{C})$*, 1. there is no query* $\mathcal{O}_{VK}(B_j)$ *with* $B_j \in Set(\vec{V}) \setminus \mathcal{C}$*; and 2. there is no query* $\mathcal{O}_{SK}(A_i)$*.*

$\mathbf{A}$*'s advantage in winning the Off-The-Record game is*

$$Adv^{\mathsf{OTR}}(\mathbf{A}) \coloneqq \left| \Pr[\mathbf{AG}_{\mathbf{0}}^{\mathsf{OTR}} = \mathtt{win}] + \Pr[\mathbf{AG}_{\mathbf{1}}^{\mathsf{OTR}} = \mathtt{win}] - 1 \right|.$$

[10] We note, however, that for technical reasons we do need an additional property from the underlying MDVS scheme. For more details refer to [12, 13].

We say that an adversary $\mathbf{A}$ (ε, t)-breaks the $(n_S, n_R, d_E, q_E, q_D)$-Correctness (resp. -Consistency, -Unforgeability, -IND-CCA-2, -IK-CCA-2, -Off-The-Record) of Π if $\mathbf{A}$ runs in time at most t, queries $\mathcal{O}_{SPK}$, $\mathcal{O}_{SK}$, $\mathcal{O}_E$ and $\mathcal{O}_D$ on at most n_S different senders, queries $\mathcal{O}_{RPK}$, $\mathcal{O}_{RK}$, $\mathcal{O}_E$ and $\mathcal{O}_D$ on at most n_R different receivers, makes at most q_E and q_D queries to $\mathcal{O}_E$ and $\mathcal{O}_D$, respectively, with the sum of lengths of the party vectors input to $\mathcal{O}_E$ being at most d_E, and $\mathbf{A}$'s advantage in winning the (corresponding) security game is at least ε. Finally, we say that Π is

$$(\varepsilon_{\mathsf{Corr}}, \varepsilon_{\mathsf{Cons}}, \varepsilon_{\mathsf{Unforg}}, \varepsilon_{\mathsf{IND\text{-}CCA\text{-}2}}, \varepsilon_{\mathsf{IK\text{-}CCA\text{-}2}}, \varepsilon_{\mathsf{OTR}}, t, n_S, n_R, d_E, q_E, q_D)\text{-secure},$$

if no adversary $\mathbf{A}$:

- $(\varepsilon_{\mathsf{Corr}}, t)$-breaks the $(n_S, n_R, d_E, q_E, q_D)$-Correctness of Π;
- $(\varepsilon_{\mathsf{Cons}}, t)$-breaks the $(n_S, n_R, d_E, q_E, q_D)$-Consistency of Π;
- $(\varepsilon_{\mathsf{Unforg}}, t)$-breaks the $(n_S, n_R, d_E, q_E, q_D)$-Unforgeability of Π;
- $(\varepsilon_{\mathsf{IND\text{-}CCA\text{-}2}}, t)$-breaks the $(n_S, n_R, d_E, q_E, q_D)$-IND-CCA-2 security of Π;
- $(\varepsilon_{\mathsf{IK\text{-}CCA\text{-}2}}, t)$-breaks the $(n_S, n_R, d_E, q_E, q_D)$-IK-CCA-2 security of Π; or
- $(\varepsilon_{\mathsf{OTR}}, t)$-breaks the $(n_S, n_R, d_E, q_E, q_D)$-Off-The-Record security of Π.

Remark 8. As one may note, due to the Off-The-Record property of MDRS-PKE schemes (see Definition 11), any receiver B_j can generate a ciphertext that decrypts correctly under B_j's own receiver secret key using only its own secret key and the public keys of the sender and any other receivers. It is thus crucial that, when defining ciphertext Unforgeability (see Definition 8), the adversary is not allowed to query for the secret key of any receiver with respect to which it is trying forge a signature.

It is equally important that the adversary is not allowed to query for the secret keys of honest receivers in the Off-The-Record security notion (Definition 11): as honest receivers do not participate in the ciphertext forgery, due to the Unforgeability of ciphertexts (Definition 8)—which in particular guarantees that if a receiver is honest, then it only decrypts ciphertexts generated by the actual sender, assuming the sender is honest—if an adversary could query for the secret key of an honest receiver B_j, it would be able to distinguish real ciphertexts generated by the sender—which B_j would decrypt successfully using its secret key—from fake ciphertexts generated by dishonest receivers—which, by the Unforgeability of ciphertexts, B_j would not decrypt successfully.

Finally, the adversary can also not be given access to the secret key of any honest receiver B_j in the Consistency game of Definition 7, as otherwise, by the Off-The-Record guarantee (Definition 11), it would be able to use B_j's receiver secret key to forge a ciphertext c that B_j would decrypt successfully (as if it really had been sent by the actual sender), whereas any other honest (designated) receiver's decryption of c would fail.

6 A Multi-Designated Receiver Signed Public Key Encryption Scheme from Standard Assumptions

In this section we show how to construct an MDRS-PKE from a PKEBC $\Pi_{\text{PKEBC}} = (S, G, E, D)$, an MDVS $\Pi_{\text{MDVS}} = (S, G_S, G_V, \mathit{Sig}, \mathit{Vfy}, \mathit{Forge})$ and a (One-Time Strongly Unforgeable) Digital Signature Scheme $\Pi_{\text{DSS}} = (G, \mathit{Sig}, \mathit{Vfy})$. The construction is simple: to encrypt one first samples a fresh DSS key-pair $(\mathtt{vk}, \mathtt{sk})$ and then uses the MDVS scheme to sign the message, the vector of PKEBC public keys of the receivers, and the verification key $\mathtt{vk}$; next, one uses the PKEBC scheme to encrypt the message, the MDVS signature, the public MDVS signer key of the sender and the vector of public MDVS verifier keys of the receivers; finally one signs the resulting (PKEBC) ciphertext using the initially sampled DSS secret key. The final ciphertext is then a triple consisting of the DSS verification key $\mathtt{vk}$, the PKEBC ciphertext c and the DSS signature σ_c on c.

Remark 9. Even though our MDRS-PKE construction allows parties to locally generate their keys, to achieve the Off-The-Record guarantee it is required that dishonest receivers know their secret keys. This is only so as otherwise one could mount attacks that break the Off-The-Record guarantee. For instance, consider an honest sender Alice that sends a message m to Bob. Bob, who is dishonest wants to convince a non-designated receiver, Eve, that Alice sent m. To do that, Bob could have Eve generating the keys for Bob herself, and give him only the public key (that Bob would claim as being his public key). When Alice sends m, Eve can now learn that Alice sent m as it can use Bob's secret key. Furthermore, since no one other than Eve has Bob's secret key, Eve knows that it cannot be a fake message, implying that it must be Alice's message. Current composable notions capturing the security of MDVS schemes solve this problem by assuming a trusted third party which generates all key-pairs and gives everyone access to their own key-pair [32][11]. This in particular implies that Bob would have access to its own secret key, and so even if Eve would know Bob's secret key, she would not be able to tell if Alice was the one sending messages or if Bob was faking Alice's messages.

6.1 Security Analysis of the MDRS-PKE Construction

Due to space restrictions, the full proofs of the following results are in the appendix (see Appendix I).

Theorem 6. *If* Π_{PKEBC} *is*

$$(\varepsilon_{\text{PKEBC-Corr}}, \varepsilon_{\text{PKEBC-Rob}}, \varepsilon_{\text{PKEBC-Cons}}, \varepsilon_{\text{PKEBC-IND-CCA-2}}, \varepsilon_{\text{PKEBC-IK-CCA-2}}, t_{\text{PKEBC}}, n_{\text{PKEBC}}, d_{E\,\text{PKEBC}}, q_{E\,\text{PKEBC}}, q_{D\,\text{PKEBC}})\text{-}\mathit{secure}, \tag{6.1}$$

[11] The composable notions capturing the security of MDVS given in [32] actually assume something even stronger: every dishonest party has access to the secret keys of every other dishonest party.

Algorithm 2 Construction of an MDRS-PKE $\Pi = (S, G_S, G_R, E, D, \mathit{Forge})$ from a PKEBC $\Pi_{\mathrm{PKEBC}} = (S, G, E, D)$, an MDVS $\Pi_{\mathrm{MDVS}} = (\mathit{Setup}, G_S, G_V, \mathit{Sign}, \mathit{Vfy}, \mathit{Forge})$, and a DSS $\Pi_{\mathrm{DSS}} = (G, \mathit{Sig}, \mathit{Vfy})$.

$S(1^k)$
  $\mathtt{pp}_{\mathrm{MDVS}} \leftarrow \Pi_{\mathrm{MDVS}}.S(1^k)$
  $\mathtt{pp}_{\mathrm{PKEBC}} \leftarrow \Pi_{\mathrm{PKEBC}}.S(1^k)$
  return $\mathtt{pp} \coloneqq (\mathtt{pp}_{\mathrm{MDVS}}, \mathtt{pp}_{\mathrm{PKEBC}}, 1^k)$

$G_S(\mathtt{pp})$
  $(\mathtt{spk}_{\mathrm{MDVS}}, \mathtt{ssk}_{\mathrm{MDVS}}) \leftarrow \Pi_{\mathrm{MDVS}}.G_S(\mathtt{pp}_{\mathrm{MDVS}})$
  return $(\mathtt{spk} \coloneqq \mathtt{spk}_{\mathrm{MDVS}}, \mathtt{ssk} \coloneqq (\mathtt{spk}, \mathtt{ssk}_{\mathrm{MDVS}}))$

$G_R(\mathtt{pp})$
  $(\mathtt{vpk}_{\mathrm{MDVS}}, \mathtt{vsk}_{\mathrm{MDVS}}) \leftarrow \Pi_{\mathrm{MDVS}}.G_V(\mathtt{pp}_{\mathrm{MDVS}})$
  $(\mathtt{pk}_{\mathrm{PKEBC}}, \mathtt{sk}_{\mathrm{PKEBC}}) \leftarrow \Pi_{\mathrm{PKEBC}}.G(\mathtt{pp}_{\mathrm{PKEBC}})$
  return $(\mathtt{rpk} \coloneqq (\mathtt{vpk}_{\mathrm{MDVS}}, \mathtt{pk}_{\mathrm{PKEBC}}), \mathtt{rsk} \coloneqq (\mathtt{rpk}, (\mathtt{vsk}_{\mathrm{MDVS}}, \mathtt{sk}_{\mathrm{PKEBC}})))$

$E_{\mathtt{pp}}(\mathtt{ssk}, \vec{v} \coloneqq (\mathtt{rpk}_1, \ldots, \mathtt{rpk}_{|\vec{v}|}), m)$
  $\vec{v}_{\mathrm{PKEBC}} \coloneqq (\mathtt{rpk}_1.\mathtt{pk}_{\mathrm{PKEBC}}, \ldots, \mathtt{rpk}_{|\vec{v}|}.\mathtt{pk}_{\mathrm{PKEBC}})$
  $\vec{v}_{\mathrm{MDVS}} \coloneqq (\mathtt{rpk}_1.\mathtt{vpk}_{\mathrm{MDVS}}, \ldots, \mathtt{rpk}_{|\vec{v}|}.\mathtt{vpk}_{\mathrm{MDVS}})$
  $(\mathtt{vk}, \mathtt{sk}) \leftarrow \Pi_{\mathrm{DSS}}.G(\mathtt{pp}.1^k)$
  $\sigma \leftarrow \Pi_{\mathrm{MDVS}}.\mathit{Sig}_{\mathtt{pp}_{\mathrm{MDVS}}}(\mathtt{ssk}_{\mathrm{MDVS}}, \vec{v}_{\mathrm{MDVS}}, (\vec{v}_{\mathrm{PKEBC}}, m, \mathtt{vk}))$
  $c \leftarrow \Pi_{\mathrm{PKEBC}}.E_{\mathtt{pp}_{\mathrm{PKEBC}}}(\vec{v}_{\mathrm{PKEBC}}, (\mathtt{spk}_{\mathrm{MDVS}}, \vec{v}_{\mathrm{MDVS}}, m, \sigma))$
  $\sigma' \leftarrow \Pi_{\mathrm{DSS}}.\mathit{Sig}_{\mathtt{sk}}(c)$
  return $(\mathtt{vk}, \sigma', c)$

$D_{\mathtt{pp}}(\mathtt{rsk}, c \coloneqq (\mathtt{vk}, \sigma', c'))$
  if $\Pi_{\mathrm{DSS}}.\mathit{Vfy}_{\mathtt{vk}}(c', \sigma') = 0$ **then**
    return $\perp$
  $(\vec{v}_{\mathrm{PKEBC}}, (\mathtt{spk} \coloneqq \mathtt{spk}_{\mathrm{MDVS}}, \vec{v}_{\mathrm{MDVS}}, m, \sigma)) \leftarrow \Pi_{\mathrm{PKEBC}}.D_{\mathtt{pp}_{\mathrm{PKEBC}}}(\mathtt{rsk}.\mathtt{sk}_{\mathrm{PKEBC}}, c')$
  if $(\vec{v}_{\mathrm{PKEBC}}, (\mathtt{spk}, \vec{v}_{\mathrm{MDVS}}, m, \sigma)) = \perp \ \vee \ |\vec{v}_{\mathrm{PKEBC}}| \neq |\vec{v}_{\mathrm{MDVS}}|$ **then**
    return $\perp$
  $\vec{v} \coloneqq ((v_{\mathrm{MDVS}1}, v_{\mathrm{PKEBC}1}), \ldots, (v_{\mathrm{MDVS}|\vec{v}_{\mathrm{PKEBC}}|}, v_{\mathrm{PKEBC}|\vec{v}_{\mathrm{PKEBC}}|}))$
  if $\mathtt{rsk}.\mathtt{rpk} \notin \vec{v}$ **then**
    return $\perp$
  if $\Pi_{\mathrm{MDVS}}.\mathit{Vfy}_{\mathtt{pp}_{\mathrm{MDVS}}}(\mathtt{spk}, \mathtt{vsk}_{\mathrm{MDVS}}, \vec{v}_{\mathrm{MDVS}}, (\vec{v}_{\mathrm{PKEBC}}, m, \mathtt{vk}), \sigma) \neq \mathtt{valid}$ **then**
    return $\perp$
  return $(\mathtt{spk}, \vec{v}, m)$

$\mathit{Forge}_{\mathtt{pp}}(\mathtt{spk}, \vec{v} \coloneqq (\mathtt{rpk}_1, \ldots, \mathtt{rpk}_{|\vec{v}|}), m, \vec{s} \coloneqq (\mathtt{rsk}_1, \ldots, \mathtt{rsk}_{|\vec{s}|}))$
  $\vec{v}_{\mathrm{PKEBC}} \coloneqq (\mathtt{rpk}_1.\mathtt{pk}_{\mathrm{PKEBC}}, \ldots, \mathtt{rpk}_{|\vec{v}|}.\mathtt{pk}_{\mathrm{PKEBC}})$
  $\vec{v}_{\mathrm{MDVS}} \coloneqq (\mathtt{rpk}_1.\mathtt{vpk}_{\mathrm{MDVS}}, \ldots, \mathtt{rpk}_{|\vec{v}|}.\mathtt{vpk}_{\mathrm{MDVS}})$
  $\vec{s}_{\mathrm{MDVS}} \coloneqq (\mathtt{rsk}_1.\mathtt{vsk}_{\mathrm{MDVS}}, \ldots, \mathtt{rsk}_{|\vec{s}|}.\mathtt{vsk}_{\mathrm{MDVS}})$
  $(\mathtt{vk}, \mathtt{sk}) \leftarrow \Pi_{\mathrm{DSS}}.G(\mathtt{pp}.1^k)$
  $\sigma \leftarrow \Pi_{\mathrm{MDVS}}.\mathit{Forge}_{\mathtt{pp}_{\mathrm{MDVS}}}(\mathtt{spk}_{\mathrm{MDVS}}, \vec{v}_{\mathrm{MDVS}}, (\vec{v}_{\mathrm{PKEBC}}, m, \mathtt{vk}), \vec{s}_{\mathrm{MDVS}})$
  $c \leftarrow \Pi_{\mathrm{PKEBC}}.E_{\mathtt{pp}_{\mathrm{PKEBC}}}(\vec{v}_{\mathrm{PKEBC}}, (\mathtt{spk}_{\mathrm{MDVS}}, \vec{v}_{\mathrm{MDVS}}, m, \sigma))$
  $\sigma' \leftarrow \Pi_{\mathrm{DSS}}.\mathit{Sig}_{\mathtt{sk}}(c)$
  return $(\mathtt{vk}, \sigma', c)$

Π_{MDVS} *is*

$$(\varepsilon_{\mathrm{MDVS\text{-}Corr}},\varepsilon_{\mathrm{MDVS\text{-}Cons}},\varepsilon_{\mathrm{MDVS\text{-}Unforg}},\varepsilon_{\mathrm{MDVS\text{-}OTR}},\ t_{\mathrm{MDVS}},n_{S\,\mathrm{MDVS}},n_{V\,\mathrm{MDVS}},d_{S\,\mathrm{MDVS}},q_{S\,\mathrm{MDVS}},q_{V\,\mathrm{MDVS}})\text{-}secure, \tag{6.2}$$

and Π_{DSS} *is*

$$(\varepsilon_{\mathrm{DSS\text{-}Corr}},\varepsilon_{\mathrm{DSS\text{-}1\text{-}sEUF\text{-}CMA}},t_{\mathrm{DSS}},n_{\mathrm{DSS}},q_{S\,\mathrm{DSS}},q_{V\,\mathrm{DSS}})\text{-}secure, \tag{6.3}$$

then no adversary $\mathbf{A}$ (ε,t)*-breaks* Π*'s*

$$\begin{aligned}&(n_S \coloneqq n_{S\,\mathrm{MDVS}},\\ &n_R \coloneqq \min(n_{\mathrm{PKEBC}},n_{V\,\mathrm{MDVS}}),\\ &d_E \coloneqq \min(d_{E\,\mathrm{PKEBC}},d_{S\,\mathrm{MDVS}}),\\ &q_E \coloneqq \min(q_{E\,\mathrm{PKEBC}},q_{S\,\mathrm{MDVS}},n_{\mathrm{DSS}},q_{S\,\mathrm{DSS}}),\\ &q_D \coloneqq \min(q_{D\,\mathrm{PKEBC}},q_{V\,\mathrm{MDVS}},q_{V\,\mathrm{DSS}}))\text{-}Correctness,\end{aligned}$$

with $\varepsilon > \varepsilon_{\mathrm{PKEBC\text{-}Corr}}+\varepsilon_{\mathrm{MDVS\text{-}Corr}}+\varepsilon_{\mathrm{DSS\text{-}Corr}}$*, and* $t_{\mathrm{PKEBC}},t_{\mathrm{MDVS}},t_{\mathrm{DSS}} \approx t+t_{\mathsf{Corr}}$*, where* t_{Corr} *is the time to run* Π*'s* $\mathbf{G}^{\mathsf{Corr}}$ *game.*

Remark 10. Similarly to Remark 3, if Π_{PKEBC}'s correctness holds even when the adversary is allowed to query for the secret key of any receiver, and Π_{MDVS}'s correctness holds even when the adversary is allowed to query for the secret keys of any signer or verifier, then Π's correctness holds even when the adversary is allowed to query for the secret keys of any sender and receiver.

Theorem 7. *If* Π_{PKEBC} *is*

$$(\varepsilon_{\mathrm{PKEBC\text{-}Corr}},\varepsilon_{\mathrm{PKEBC\text{-}Rob}},\varepsilon_{\mathrm{PKEBC\text{-}Cons}},\varepsilon_{\mathrm{PKEBC\text{-}IND\text{-}CCA\text{-}2}},\varepsilon_{\mathrm{PKEBC\text{-}IK\text{-}CCA\text{-}2}},\ t_{\mathrm{PKEBC}},n_{\mathrm{PKEBC}},d_{E\,\mathrm{PKEBC}},q_{E\,\mathrm{PKEBC}},q_{D\,\mathrm{PKEBC}})\text{-}secure, \tag{6.4}$$

Π_{MDVS} *is*

$$(\varepsilon_{\mathrm{MDVS\text{-}Corr}},\varepsilon_{\mathrm{MDVS\text{-}Cons}},\varepsilon_{\mathrm{MDVS\text{-}Unforg}},\varepsilon_{\mathrm{MDVS\text{-}OTR}},\ t_{\mathrm{MDVS}},n_{S\,\mathrm{MDVS}},n_{V\,\mathrm{MDVS}},d_{S\,\mathrm{MDVS}},q_{S\,\mathrm{MDVS}},q_{V\,\mathrm{MDVS}})\text{-}secure, \tag{6.5}$$

Π_{DSS} *is*

$$(\varepsilon_{\mathrm{DSS\text{-}Corr}},\varepsilon_{\mathrm{DSS\text{-}1\text{-}EUF\text{-}CMA}},t_{\mathrm{DSS}},n_{\mathrm{DSS}},q_{S\,\mathrm{DSS}},q_{V\,\mathrm{DSS}})\text{-}secure, \tag{6.6}$$

and $\Pi_{\mathrm{DSS}}.Vfy$ *is a deterministic algorithm, then no adversary* $\mathbf{A}$ (ε,t)*-breaks* Π*'s*

$$\begin{aligned}&(n_S \coloneqq n_{S\,\mathrm{MDVS}},n_R \coloneqq \min(n_{\mathrm{PKEBC}},n_{V\,\mathrm{MDVS}}),\\ &d_E \coloneqq \min(d_{E\,\mathrm{PKEBC}},d_{S\,\mathrm{MDVS}}),q_E \coloneqq \min(q_{E\,\mathrm{PKEBC}},q_{S\,\mathrm{MDVS}}),\\ &q_D \coloneqq \min(q_{D\,\mathrm{PKEBC}},q_{V\,\mathrm{MDVS}}))\text{-}Consistency,\end{aligned}$$

with $\varepsilon > \varepsilon_{\mathrm{PKEBC\text{-}Cons}}+\varepsilon_{\mathrm{MDVS\text{-}Cons}}$*, and* $t_{\mathrm{PKEBC}},t_{\mathrm{MDVS}} \approx t+t_{\mathsf{Cons}}$*, where* t_{Cons} *is the time to run* Π*'s* $\mathbf{G}^{\mathsf{Cons}}$ *game.*

Theorem 8. *If* Π_{MDVS} *is*

$$\begin{aligned}(&\varepsilon_{\text{MDVS-Corr}},\varepsilon_{\text{MDVS-Cons}},\varepsilon_{\text{MDVS-Unforg}},\varepsilon_{\text{MDVS-OTR}},\\ &t_{\mathrm{MDVS}},n_{S\,\mathrm{MDVS}},n_{V\,\mathrm{MDVS}},d_{S\,\mathrm{MDVS}},q_{S\,\mathrm{MDVS}},q_{V\,\mathrm{MDVS}})\text{-}\textit{secure},\end{aligned} \tag{6.7}$$

and Π_{DSS} *is*

$$(\varepsilon_{\text{DSS-Corr}},\varepsilon_{\text{DSS-1-EUF-CMA}},t_{\mathrm{DSS}},n_{\mathrm{DSS}},q_{S\,\mathrm{DSS}},q_{V\,\mathrm{DSS}})\text{-}\textit{secure}, \tag{6.8}$$

then no adversary $\mathbf{A}$ (ε,t)*-breaks* Π*'s*

$$\begin{aligned}&(n_S \coloneqq n_{S\,\mathrm{MDVS}}, n_R \coloneqq n_{V\,\mathrm{MDVS}},\\ &d_E \coloneqq d_{S\,\mathrm{MDVS}}, q_E \coloneqq \min(q_{S\,\mathrm{MDVS}}, n_{\mathrm{DSS}}, q_{S\,\mathrm{DSS}}),\\ &q_D \coloneqq \min(q_{V\,\mathrm{MDVS}}, q_{V\,\mathrm{DSS}}))\text{-}\textit{Unforgeability},\end{aligned}$$

with $\varepsilon > \varepsilon_{\text{DSS-1-sEUF-CMA}} + \varepsilon_{\text{MDVS-Unforg}}$*, and* $t_{\mathrm{DSS}}, t_{\mathrm{MDVS}} \approx t + t_{\mathsf{Unforg}}$*, where* t_{Unforg} *is the time to run* Π*'s* $\mathbf{G}^{\mathsf{Unforg}}$ *game.*

Theorem 9. *If* Π_{PKEBC} *is*

$$\begin{aligned}(&\varepsilon_{\text{PKEBC-Corr}},\varepsilon_{\text{PKEBC-Rob}},\varepsilon_{\text{PKEBC-Cons}},\varepsilon_{\text{PKEBC-IND-CCA-2}},\varepsilon_{\text{PKEBC-IK-CCA-2}},\\ &t_{\mathrm{PKEBC}},n_{\mathrm{PKEBC}},d_{E\,\mathrm{PKEBC}},q_{E\,\mathrm{PKEBC}},q_{D\,\mathrm{PKEBC}})\text{-}\textit{secure},\end{aligned} \tag{6.9}$$

Π_{MDVS} *is*

$$\begin{aligned}(&\varepsilon_{\text{MDVS-Corr}},\varepsilon_{\text{MDVS-Cons}},\varepsilon_{\text{MDVS-Unforg}},\varepsilon_{\text{MDVS-OTR}},\\ &t_{\mathrm{MDVS}},n_{S\,\mathrm{MDVS}},n_{V\,\mathrm{MDVS}},d_{S\,\mathrm{MDVS}},q_{S\,\mathrm{MDVS}},q_{V\,\mathrm{MDVS}})\text{-}\textit{secure},\end{aligned} \tag{6.10}$$

and Π_{DSS} *is*

$$(\varepsilon_{\text{DSS-Corr}},\varepsilon_{\text{DSS-1-sEUF-CMA}},t_{\mathrm{DSS}},n_{\mathrm{DSS}},q_{S\,\mathrm{DSS}},q_{V\,\mathrm{DSS}})\text{-}\textit{secure}, \tag{6.11}$$

then no adversary $\mathbf{A}$ (ε,t)*-breaks* Π*'s*

$$\begin{aligned}\Big(&n_S \coloneqq n_{S\,\mathrm{MDVS}},\\ &n_R \coloneqq \min(n_{\mathrm{PKEBC}}, n_{V\,\mathrm{MDVS}}),\\ &d_E \coloneqq \min(d_{E\,\mathrm{PKEBC}}, d_{S\,\mathrm{MDVS}}),\\ &q_E \coloneqq \min(q_{E\,\mathrm{PKEBC}}, q_{S\,\mathrm{MDVS}}, n_{\mathrm{DSS}}, q_{S\,\mathrm{DSS}}),\\ &q_D \coloneqq \min(q_{D\,\mathrm{PKEBC}}, q_{V\,\mathrm{MDVS}}, q_{V\,\mathrm{DSS}})\Big)\text{-IND-CCA-2 }\textit{security},\end{aligned}$$

with

$$\begin{aligned}\varepsilon > 2 \cdot (&\varepsilon_{\text{DSS-1-EUF-CMA}} + \varepsilon_{\text{MDVS-Unforg}} + \varepsilon_{\text{PKEBC-Rob}})\\ &+ 4 \cdot \varepsilon_{\text{PKEBC-Corr}} + \varepsilon_{\text{PKEBC-IND-CCA-2}},\end{aligned}$$

and with $t_{\mathrm{DSS}}, t_{\mathrm{MDVS}}, t_{\mathrm{PKEBC}} \approx t + t_{\mathsf{IND\text{-}CCA\text{-}2}}$*, where* $t_{\mathsf{IND\text{-}CCA\text{-}2}}$ *is the time to run* Π*'s* $\mathbf{G}^{\mathsf{IND\text{-}CCA\text{-}2}}$ *games.*

Remark 11. Note that Definitions 9 and 10 do not allow an adversary to query for the secret keys of any sender A_i that is given as input to a query to $\mathcal{O}_E$. Furthermore, we do not know how such a reduction could go through without requiring any additional properties from the underlying MDVS scheme. In [12,13], Chakraborty et al. introduce Bounded Message Validity and prove the IND-CCA-2 and IK-CCA-2 security of the MDRS-PKE scheme we give in this paper (Algorithm 2). We refer the interested reader to [13] for further details.

Theorem 10. *If* Π_{PKEBC} *is*

$$\begin{aligned}(&\varepsilon_{\text{PKEBC-Corr}},\varepsilon_{\text{PKEBC-Rob}},\varepsilon_{\text{PKEBC-Cons}},\varepsilon_{\text{PKEBC-IND-CCA-2}},\varepsilon_{\text{PKEBC-IK-CCA-2}},\\ &t_{\text{PKEBC}},n_{\text{PKEBC}},d_{E\,\text{PKEBC}},q_{E\,\text{PKEBC}},q_{D\,\text{PKEBC}})\text{-}\textit{secure,}\end{aligned} \tag{6.12}$$

Π_{MDVS} *is*

$$\begin{aligned}(&\varepsilon_{\text{MDVS-Corr}},\varepsilon_{\text{MDVS-Cons}},\varepsilon_{\text{MDVS-Unforg}},\varepsilon_{\text{MDVS-OTR}},\\ &t_{\text{MDVS}},n_{S\,\text{MDVS}},n_{V\,\text{MDVS}},d_{S\,\text{MDVS}},q_{S\,\text{MDVS}},q_{V\,\text{MDVS}})\text{-}\textit{secure,}\end{aligned} \tag{6.13}$$

and Π_{DSS} *is*

$$(\varepsilon_{\text{DSS-Corr}},\varepsilon_{\text{DSS-1-sEUF-CMA}},t_{\text{DSS}},n_{\text{DSS}},q_{S\,\text{DSS}},q_{V\,\text{DSS}})\text{-}\textit{secure,} \tag{6.14}$$

then no adversary $\mathbf{A}$ (ε,t)*-breaks* Π*'s*

$$\begin{aligned}\Big(&n_S \coloneqq n_{S\,\text{MDVS}},\\ &n_R \coloneqq \min(n_{\text{PKEBC}},n_{V\,\text{MDVS}}),\\ &d_E \coloneqq \min(d_{E\,\text{PKEBC}},d_{S\,\text{MDVS}}),\\ &q_E \coloneqq \min(q_{E\,\text{PKEBC}},q_{S\,\text{MDVS}},n_{\text{DSS}},q_{S\,\text{DSS}}),\\ &q_D \coloneqq \min(q_{D\,\text{PKEBC}},q_{V\,\text{MDVS}},q_{V\,\text{DSS}})\Big)\text{-IK-CCA-2 }\textit{security,}\end{aligned}$$

with

$$\begin{aligned}\varepsilon > 2\cdot(&\varepsilon_{\text{DSS-1-EUF-CMA}}+\varepsilon_{\text{MDVS-Unforg}}+\varepsilon_{\text{PKEBC-Rob}})\\ &+4\cdot\varepsilon_{\text{PKEBC-Corr}}+\varepsilon_{\text{PKEBC-IK-CCA-2}}+\varepsilon_{\text{PKEBC-IND-CCA-2}},\end{aligned}$$

and with $t_{\text{DSS}},t_{\text{MDVS}},t_{\text{PKEBC}} \approx t + t_{\text{IK-CCA-2}}$*, where* $t_{\text{IK-CCA-2}}$ *is the time to run* Π*'s* $\mathbf{G}^{\text{IK-CCA-2}}$ *games.*

Theorem 11. *If* Π_{MDVS} *is*

$$\begin{aligned}(&\varepsilon_{\text{MDVS-Corr}},\varepsilon_{\text{MDVS-Cons}},\varepsilon_{\text{MDVS-Unforg}},\varepsilon_{\text{MDVS-OTR}},\\ &t_{\text{MDVS}},n_{S\,\text{MDVS}},n_{V\,\text{MDVS}},d_{S\,\text{MDVS}},q_{S\,\text{MDVS}},q_{V\,\text{MDVS}})\text{-}\textit{secure,}\end{aligned} \tag{6.15}$$

and Π_{DSS} *is*

$$(\varepsilon_{\text{DSS-Corr}},\varepsilon_{\text{DSS-1-EUF-CMA}},t_{\text{DSS}},n_{\text{DSS}},q_{S\,\text{DSS}},q_{V\,\text{DSS}})\text{-}\textit{secure,} \tag{6.16}$$

then no adversary $\mathbf{A}$ (ε, t)*-breaks* Π*'s*

$$\begin{aligned}&(n_S \coloneqq n_{S\,\mathrm{MDVS}}, n_R \coloneqq n_{V\,\mathrm{MDVS}},\\ &d_E \coloneqq d_{S\,\mathrm{MDVS}}, q_E \coloneqq \min(q_{S\,\mathrm{MDVS}}, n_{\mathrm{DSS}}, q_{S\,\mathrm{DSS}}),\\ &q_D \coloneqq \min(q_{V\,\mathrm{MDVS}}, q_{V\,\mathrm{DSS}}))\text{-}\textit{Off-The-Record security,}\end{aligned}$$

with $\varepsilon > 2 \cdot \varepsilon_{\text{DSS-1-sEUF-CMA}} + \varepsilon_{\text{MDVS-OTR}}$*, and* $t_{\mathrm{DSS}}, t_{\mathrm{MDVS}} \approx t + t_{\mathsf{OTR}}$*, where* t_{OTR} *is the time to run* Π*'s* $\mathbf{G}^{\mathsf{OTR}}$ *games.*

Remark 12. It is easy to see from the proof of Theorem 11 that if Π_{MDVS} satisfies a stronger Off-The-Record notion in which the adversary is allowed to query for the secret key of any sender, then Π would also satisfy the analogous stronger Off-The-Record notion for MDRS-PKE schemes in which the adversary is allowed to query for the secret key of any sender.

7 Acknowledgments

The authors would like to thank Dennis Hofheinz for helpful discussions, for suggesting Naor-Yung's scheme [35] together with a Simulation Sound NIZK scheme [37] and a Binding Commitment scheme as a starting point to construct the PKEBC scheme, and for suggesting the use of a Digital Signature Scheme scheme in the MDRS-PKE construction (to tie MDVS signatures and PKEBC ciphertexts together). The authors would also like to thank Christian Badertscher, Daniel Jost and Chen-Da Liu-Zhang for helpful discussions.

References

1. Abdalla, M., Bellare, M., Neven, G.: Robust encryption. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 480–497. Springer, Heidelberg (Feb 2010). https://doi.org/10.1007/978-3-642-11799-2˙28
2. Alwen, J., Coretti, S., Dodis, Y., Tselekounis, Y.: Security analysis and improvements for the IETF MLS standard for group messaging. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part I. LNCS, vol. 12170, pp. 248–277. Springer, Heidelberg (Aug 2020). https://doi.org/10.1007/978-3-030-56784-2˙9
3. An, J.H., Dodis, Y., Rabin, T.: On the security of joint signature and encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 83–107. Springer, Heidelberg (Apr / May 2002). https://doi.org/10.1007/3-540-46035-7˙6
4. Badertscher, C., Banfi, F., Maurer, U.: A constructive perspective on signcryption security. In: Catalano, D., De Prisco, R. (eds.) SCN 18. LNCS, vol. 11035, pp. 102–120. Springer, Heidelberg (Sep 2018). https://doi.org/10.1007/978-3-319-98113-0˙6
5. Badertscher, C., Maurer, U., Portmann, C., Rito, G.: Revisiting (R)CCA security and replay protection. In: Garay, J. (ed.) PKC 2021, Part II. LNCS, vol. 12711, pp. 173–202. Springer, Heidelberg (May 2021). https://doi.org/10.1007/978-3-030-75248-4˙7
6. Barth, A., Boneh, D., Waters, B.: Privacy in encrypted content distribution using private broadcast encryption. In: Di Crescenzo, G., Rubin, A. (eds.) FC 2006. LNCS, vol. 4107, pp. 52–64. Springer, Heidelberg (Feb / Mar 2006)

7. Bellare, M., Boldyreva, A., Desai, A., Pointcheval, D.: Key-privacy in public-key encryption. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 566–582. Springer, Heidelberg (Dec 2001). https://doi.org/10.1007/3-540-45682-1˙33
8. Bellare, M., Boldyreva, A., Micali, S.: Public-key encryption in a multi-user setting: Security proofs and improvements. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 259–274. Springer, Heidelberg (May 2000). https://doi.org/10.1007/3-540-45539-6˙18
9. Bellare, M., Rogaway, P.: The security of triple encryption and a framework for code-based game-playing proofs. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 409–426. Springer, Heidelberg (May / Jun 2006). https://doi.org/10.1007/11761679˙25
10. Boneh, D., Gentry, C., Waters, B.: Collusion resistant broadcast encryption with short ciphertexts and private keys. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 258–275. Springer, Heidelberg (Aug 2005). https://doi.org/10.1007/11535218˙16
11. Borisov, N., Goldberg, I., Brewer, E.A.: Off-the-record communication, or, why not to use PGP. In: Atluri, V., Syverson, P.F., di Vimercati, S.D.C. (eds.) WPES 2004. pp. 77–84. ACM (2004), https://doi.org/10.1145/1029179.1029200
12. Chakraborty, S., Hofheinz, D., Maurer, U., Rito, G.: Deniable authentication when signing keys leak. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part III. LNCS, vol. 14006, pp. 69–100. Springer, Heidelberg (Apr 2023). https://doi.org/10.1007/978-3-031-30620-4˙3
13. Chakraborty, S., Hofheinz, D., Maurer, U., Rito, G.: Deniable authentication when signing keys leak. Cryptology ePrint Archive, Report 2023/213 (2023), https://eprint.iacr.org/2023/213
14. Chow, S.S.M., Franklin, M.K., Zhang, H.: Practical dual-receiver encryption - soundness, complete non-malleability, and applications. In: Benaloh, J. (ed.) CT-RSA 2014. LNCS, vol. 8366, pp. 85–105. Springer, Heidelberg (Feb 2014). https://doi.org/10.1007/978-3-319-04852-9˙5
15. Cohn-Gordon, K., Cremers, C., Dowling, B., Garratt, L., Stebila, D.: A formal security analysis of the signal messaging protocol. Journal of Cryptology **33**(4), 1914–1983 (Oct 2020). https://doi.org/10.1007/s00145-020-09360-1
16. Damgård, I., Haagh, H., Mercer, R., Nitulescu, A., Orlandi, C., Yakoubov, S.: Stronger security and constructions of multi-designated verifier signatures. In: Pass, R., Pietrzak, K. (eds.) TCC 2020, Part II. LNCS, vol. 12551, pp. 229–260. Springer, Heidelberg (Nov 2020). https://doi.org/10.1007/978-3-030-64378-2˙9
17. Diament, T., Lee, H.K., Keromytis, A.D., Yung, M.: The dual receiver cryptosystem and its applications. In: Atluri, V., Pfitzmann, B., McDaniel, P. (eds.) ACM CCS 2004. pp. 330–343. ACM Press (Oct 2004). https://doi.org/10.1145/1030083.1030128
18. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO'84. LNCS, vol. 196, pp. 10–18. Springer, Heidelberg (Aug 1984)
19. Fiat, A., Naor, M.: Broadcast encryption. In: Stinson, D.R. (ed.) CRYPTO'93. LNCS, vol. 773, pp. 480–491. Springer, Heidelberg (Aug 1994). https://doi.org/10.1007/3-540-48329-2˙40
20. Gay, R., Hofheinz, D., Kiltz, E., Wee, H.: Tightly CCA-secure encryption without pairings. In: Fischlin, M., Coron, J.S. (eds.) EUROCRYPT 2016, Part I. LNCS, vol. 9665, pp. 1–27. Springer, Heidelberg (May 2016). https://doi.org/10.1007/978-3-662-49890-3˙1
21. Gegier, K.: On novel constructions of dual receiver key encapsulation mechanisms based on deterministic encryption. (2020)

22. Gjøsteen, K., Kråkmo, L.: Universally composable signcryption. In: López, J., Samarati, P., Ferrer, J.L. (eds.) EuroPKI 2007. LNCS, vol. 4582, pp. 346–353. Springer (2007), https://doi.org/10.1007/978-3-540-73408-6_26
23. Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences **28**(2), 270–299 (1984)
24. Jakobsson, M., Sako, K., Impagliazzo, R.: Designated verifier proofs and their applications. In: Maurer, U.M. (ed.) EUROCRYPT'96. LNCS, vol. 1070, pp. 143–154. Springer, Heidelberg (May 1996). https://doi.org/10.1007/3-540-68339-9˙13
25. Jost, D., Maurer, U., Mularczyk, M.: Efficient ratcheting: Almost-optimal guarantees for secure messaging. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part I. LNCS, vol. 11476, pp. 159–188. Springer, Heidelberg (May 2019). https://doi.org/10.1007/978-3-030-17653-2˙6
26. Kohlweiss, M., Maurer, U., Onete, C., Tackmann, B., Venturi, D.: Anonymity-preserving public-key encryption: A constructive approach. In: De Cristofaro, E., Wright, M.K. (eds.) PETS 2013. LNCS, vol. 7981, pp. 19–39. Springer, Heidelberg (Jul 2013). https://doi.org/10.1007/978-3-642-39077-7˙2
27. Laguillaumie, F., Vergnaud, D.: Multi-designated verifiers signatures. In: López, J., Qing, S., Okamoto, E. (eds.) ICICS 04. LNCS, vol. 3269, pp. 495–507. Springer, Heidelberg (Oct 2004)
28. Laguillaumie, F., Vergnaud, D.: Designated verifier signatures: Anonymity and efficient construction from any bilinear map. In: Blundo, C., Cimato, S. (eds.) SCN 04. LNCS, vol. 3352, pp. 105–119. Springer, Heidelberg (Sep 2005). https://doi.org/10.1007/978-3-540-30598-9˙8
29. Li, Y., Susilo, W., Mu, Y., Pei, D.: Designated verifier signature: Definition, framework and new constructions. In: Indulska, J., Ma, J., Yang, L.T., Ungerer, T., Cao, J. (eds.) UIC 2007. LNCS, vol. 4611, pp. 1191–1200. Springer (2007), https://doi.org/10.1007/978-3-540-73549-6_116
30. Libert, B., Paterson, K.G., Quaglia, E.A.: Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 206–224. Springer, Heidelberg (May 2012). https://doi.org/10.1007/978-3-642-30057-8˙13
31. Lipmaa, H., Wang, G., Bao, F.: Designated verifier signature schemes: Attacks, new security notions and a new construction. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 459–471. Springer, Heidelberg (Jul 2005). https://doi.org/10.1007/11523468˙38
32. Maurer, U., Portmann, C., Rito, G.: Giving an adversary guarantees (or: How to model designated verifier signatures in a composable framework). In: Tibouchi, M., Wang, H. (eds.) ASIACRYPT 2021, Part III. LNCS, vol. 13092, pp. 189–219. Springer, Heidelberg (Dec 2021). https://doi.org/10.1007/978-3-030-92078-4˙7
33. Maurer, U., Portmann, C., Rito, G.: Multi-designated receiver signed public key encryption. In: Dunkelman, O., Dziembowski, S. (eds.) EUROCRYPT 2022, Part II. LNCS, vol. 13276, pp. 644–673. Springer, Heidelberg (May / Jun 2022). https://doi.org/10.1007/978-3-031-07085-3˙22
34. Maurer, U., Portmann, C., Rito, G.: Multi-designated receiver signed public key encryption. Cryptology ePrint Archive, Report 2022/256 (2022), https://eprint.iacr.org/2022/256
35. Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: 22nd ACM STOC. pp. 427–437. ACM Press (May 1990). https://doi.org/10.1145/100216.100273

36. Rackoff, C., Simon, D.R.: Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In: Feigenbaum, J. (ed.) CRYPTO'91. LNCS, vol. 576, pp. 433–444. Springer, Heidelberg (Aug 1992). https://doi.org/10.1007/3-540-46766-1˙35
37. Sahai, A.: Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In: 40th FOCS. pp. 543–553. IEEE Computer Society Press (Oct 1999). https://doi.org/10.1109/SFFCS.1999.814628
38. Shoup, V.: Sequences of games: a tool for taming complexity in security proofs. Cryptology ePrint Archive, Report 2004/332 (2004), https://eprint.iacr.org/2004/332
39. Steinfeld, R., Bull, L., Wang, H., Pieprzyk, J.: Universal designated-verifier signatures. In: Laih, C.S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 523–542. Springer, Heidelberg (Nov / Dec 2003). https://doi.org/10.1007/978-3-540-40061-5˙33
40. Steinfeld, R., Wang, H., Pieprzyk, J.: Efficient extension of standard Schnorr/RSA signatures into universal designated-verifier signatures. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 86–100. Springer, Heidelberg (Mar 2004). https://doi.org/10.1007/978-3-540-24632-9˙7
41. Zhang, Y., Au, M.H., Yang, G., Susilo, W.: (strong) multi-designated verifiers signatures secure against rogue key attack. In: Xu, L., Bertino, E., Mu, Y. (eds.) NSS 2012. LNCS, vol. 7645, pp. 334–347. Springer (2012), https://doi.org/10.1007/978-3-642-34601-9_25

Appendix

A Game-Based Security Definitions for Public Key Encryption Schemes

A Public Key Encryption (PKE) scheme Π with message space $\mathcal{M}$ is a triple of PPTs $\Pi = (G, E, D)$. Below we state the Correctness and the multi-user multi-challenge variants of IND-CPA and IK-CPA security for PKE schemes (first introduced in [23] and [7], respectively). Throughout the rest of this section, let $\Pi = (G, E, D)$ be a PKE scheme. As before, we assume the game systems of the following definitions have (an implicitly defined) security parameter k.

Definition 12. *A* PKE *scheme $\Pi = (G, E, D)$ with message space $\mathcal{M}$ is perfectly correct if:*

$$\forall m \in \mathcal{M} \quad \forall(\mathtt{pk}, \mathtt{sk}) \in \mathit{Supp}(\Pi.G(1^k)) \quad \Pr[\Pi.D(\mathtt{sk}, \Pi.D(\mathtt{pk}, m)) = m] = 1.$$

Definitions 13 and 14 provide adversaries with access to oracle $\mathcal{O}_{PK}$:

Public Key Generation Oracle: $\mathcal{O}_{PK}(B_j)$

1. On the first call on B_j, compute and store $(\mathtt{pk}_j, \mathtt{sk}_j) \leftarrow G(1^k)$; output $\mathtt{pk}_j$;
2. On subsequent calls, simply output $\mathtt{pk}_j$.

The IND-CPA game systems provide adversaries with access to the additional oracle $\mathcal{O}_E$ defined below:

Encryption Oracle: $\mathcal{O}_E(B_j, m_0, m_1)$

1. For game system $\mathbf{G}_{\mathbf{b}}^{\mathsf{IND\text{-}CPA}}$, the oracle encrypts $m_{\mathbf{b}}$ under B_j's public key, $\mathtt{pk}_j$, creating a fresh ciphertext c;
2. The oracle outputs the resulting ciphertext c back to the adversary.

Definition 13. *For $\mathbf{b} \in \{0, 1\}$, consider the following game played between an adversary $\mathbf{A}$ and game system $\mathbf{G}_{\mathbf{b}}^{\mathsf{IND\text{-}CPA}}$:*

– $b' \leftarrow \mathbf{A}^{\mathcal{O}_{PK}, \mathcal{O}_E}$

$\mathbf{A}$ wins the game if $b' = \mathbf{b}$ and for every query $\mathcal{O}_E(B_j, m_0, m_1)$, $|m_0| = |m_1|$.

We define the advantage of $\mathbf{A}$ in winning the IND-CPA *security game as*

$$\mathit{Adv}^{\mathsf{IND\text{-}CPA}}(\mathbf{A}) \coloneqq \Big|\Pr[\mathbf{AG}_{\mathbf{0}}^{\mathsf{IND\text{-}CPA}} = \mathtt{win}] + \Pr[\mathbf{AG}_{\mathbf{1}}^{\mathsf{IND\text{-}CPA}} = \mathtt{win}] - 1\Big|.$$

Similarly to the IND-CPA game systems, the IK-CPA game systems provide adversaries with access to oracle $\mathcal{O}_{PK}$ and to an oracle $\mathcal{O}_E$ which behaves as follows:

Encryption Oracle: $\mathcal{O}_E(B_{j,0}, B_{j,1}, m)$

1. For game system $\mathbf{G}_{\mathbf{b}}^{\mathsf{IK\text{-}CPA}}$, encrypt m under $B_{j,\mathbf{b}}$'s public key, $\mathtt{pk}_{j,\mathbf{b}}$, creating a fresh ciphertext c;
2. Output the resulting ciphertext c back to the adversary.

Definition 14. *Consider the following game played between an adversary* $\mathbf{A}$ *and game system* $\mathbf{G}_{\mathbf{b}}^{\mathsf{IK\text{-}CPA}}$*, with* $\mathbf{b} \in \{0,1\}$*:*

– $b' \leftarrow \mathbf{A}^{\mathcal{O}_{PK},\mathcal{O}_E}$

$\mathbf{A}$ *wins the game if* $b' = \mathbf{b}$.

We define the advantage of $\mathbf{A}$ *in winning the* IK-CPA *security game as*

$$Adv^{\mathsf{IK\text{-}CPA}}(\mathbf{A}) := \Big|\Pr[\mathbf{AG}_{\mathbf{0}}^{\mathsf{IK\text{-}CPA}} = \mathtt{win}] + \Pr[\mathbf{AG}_{\mathbf{1}}^{\mathsf{IK\text{-}CPA}} = \mathtt{win}] - 1\Big|.$$

We say $\mathbf{A}$ $(\varepsilon_{\mathsf{IND\text{-}CPA}}, t)$-breaks (resp. $(\varepsilon_{\mathsf{IK\text{-}CPA}}, t)$-breaks) the (n, q_E)-IND-CPA (resp. (n, q_E)-IK-CPA) security of a PKE scheme Π if $\mathbf{A}$ runs in time at most t, queries the oracles it has access to on at most n different parties, makes at most q_E queries to oracle $\mathcal{O}_E$, and satisfies $Adv^{\mathsf{IND\text{-}CPA}}(\mathbf{A}) \geq \varepsilon_{\mathsf{IND\text{-}CPA}}$ (resp. $Adv^{\mathsf{IK\text{-}CPA}}(\mathbf{A}) \geq \varepsilon_{\mathsf{IK\text{-}CPA}}$).

Finally, we say that Π is $(\varepsilon_{\mathsf{IND\text{-}CPA}}, \varepsilon_{\mathsf{IK\text{-}CPA}}, t, n, q_E)$-secure if it is correct and no adversary $\mathbf{A}$ $(\varepsilon_{\mathsf{IND\text{-}CPA}}, t)$-breaks the (n, q_E)-IND-CPA security of Π, or $(\varepsilon_{\mathsf{IK\text{-}CPA}}, t)$-breaks the (n, q_E)-IK-CPA security of Π.

B Game-Based Security Definitions for Digital Signature Schemes

A Digital Signature Scheme (DSS) Π for a message space $\mathcal{M}$ is a triple $\Pi = (G, Sig, Vfy)$ of PPTs. Below we state the definitions of Correctness and (One-Time) Strong Existential Unforgeability under Chosen Message Attacks for DSS schemes. The notions ahead make use of oracles $\mathcal{O}_{VK}$, $\mathcal{O}_S$ and $\mathcal{O}_V$, which, for a DSS $\Pi = (G, Sig, Vfy)$, are defined as:

Key-Pair Generation Oracle: $\mathcal{O}_{VK}(i \in \mathbb{N})$

1. On the first query on i, compute and store $(\mathtt{vk}_i, \mathtt{sk}_i) \leftarrow G(1^k)$;
2. Output $\mathtt{vk}_i$.

Signing Oracle: $\mathcal{O}_S(i, m)$

1. Compute $\sigma \leftarrow Sig_{\mathtt{sk}_i}(m)$, where $\mathtt{sk}_i$ is the signing key associated with i; output σ.

Verification Oracle: $\mathcal{O}_V(i, m, \sigma)$

1. Compute $d \leftarrow Vfy_{\mathtt{vk}_i}(m, \sigma)$, where $\mathtt{vk}_i$ is the verification key associated with i; output d.

Definition 15. *Consider the following game played between an adversary* $\mathbf{A}$ *and game system* $\mathbf{G}^{\mathsf{Corr}}$*:*

1. $\mathbf{A}^{\mathcal{O}_{VK},\mathcal{O}_S,\mathcal{O}_V}$

$\mathbf{A}$ *wins if there are two queries* q_S *and* q_V *to* $\mathcal{O}_S$ *and* $\mathcal{O}_V$*, respectively, where* q_S *has input* (i, m) *and* q_V *has (matching) input* (i, m, σ)*—*σ *being the output of* q_S*—and the output of* q_V *is* 0*.*

We define the advantage of $\mathbf{A}$ *in winning the correctness game as*

$$Adv^{\mathsf{Corr}}(\mathbf{A}) \coloneqq \Pr[\mathbf{AG}^{\mathsf{Corr}} = \mathtt{win}].$$

Definition 16. *Consider the following game played between adversary* $\mathbf{A}$ *and game system* $\mathbf{G}^{1\text{-}\mathsf{sEUF\text{-}CMA}}$*:*

1. $\mathbf{A}^{\mathcal{O}_{VK},\mathcal{O}_S,\mathcal{O}_V}$

$\mathbf{A}$ *wins the game if there is a query to* $\mathcal{O}_V$ *on some input* (i^*, m^*, σ^*) *that outputs* 1*, there is no query to* $\mathcal{O}_S$ *on input* (i^*, m^*) *that output* σ^**, and for each* $i \in \mathbb{N}$ *there is only at most one query to* $\mathcal{O}_S$ *with input* i*.*

The advantage of adversary $\mathbf{A}$ *in winning game* $\mathbf{G}^{1\text{-}\mathsf{sEUF\text{-}CMA}}$ *is*

$$Adv^{1\text{-}\mathsf{sEUF\text{-}CMA}}(\mathbf{A}) \coloneqq \Pr[\mathbf{AG}^{1\text{-}\mathsf{sEUF\text{-}CMA}} = \mathtt{win}].$$

An adversary $\mathbf{A}$ $(\varepsilon_{\mathsf{Corr}}, t)$-breaks the (n, q_S, q_V)-Correctness of a DSS Π if it runs in time t, queries $\mathcal{O}_{VK}$, $\mathcal{O}_S$ and $\mathcal{O}_V$ on at most n different indices, makes at most q_S and q_V queries to oracles $\mathcal{O}_S$ and $\mathcal{O}_V$, respectively, and satisfies $Adv^{\mathsf{Corr}}(\mathbf{A}) \geq \varepsilon_{\mathsf{Corr}}$. Similarly, $\mathbf{A}$ $(\varepsilon_{1\text{-}\mathsf{sEUF\text{-}CMA}}, t)$-breaks the (n, q_S, q_V)-1-sEUF-CMA security of Π if $\mathbf{A}$ runs in time at most t, queries $\mathcal{O}_{VK}$, $\mathcal{O}_S$ and $\mathcal{O}_V$ on at most n different indices, makes at most q_S and q_V queries to, respectively, $\mathcal{O}_S$ and $\mathcal{O}_V$, and satisfies $Adv^{1\text{-}\mathsf{sEUF\text{-}CMA}}(\mathbf{A}) \geq \varepsilon_{1\text{-}\mathsf{sEUF\text{-}CMA}}$. Finally, Π is $(\varepsilon_{\mathsf{Corr}}, \varepsilon_{1\text{-}\mathsf{sEUF\text{-}CMA}}, t, n, q_S, q_V)$-secure if no adversary $\mathbf{A}$ $(\varepsilon_{\mathsf{Corr}}, t)$-breaks the (n, q_S, q_V)-Correctness of Π, or $(\varepsilon_{1\text{-}\mathsf{sEUF\text{-}CMA}}, t)$-breaks the (n, q_S, q_V)-1-sEUF-CMA security of Π.

C Game-Based Security Definitions for Binding Commitment Schemes

A Commitment Scheme (CS) for a message space $\mathcal{M}$ is a protocol consisting of a pair of PPT algorithms $\Pi = (G_{CRS}, \mathit{Commit})$. We now move to introduce game-based notions capturing the security of CS protocols. We assume the game systems ahead have (an implicitly defined) security parameter k.

The game systems of Definitions 17 and 18 provide adversaries with access to an oracle $\mathcal{O}_S$, defined as:

CRS Generation Oracle: $\mathcal{O}_S$

1. On the first call, compute and store $\mathtt{crs} \leftarrow G_{CRS}(1^k)$; output $\mathtt{crs}$;

2. On subsequent calls, output the previously generated $\texttt{crs}$.

Definition 17 captures the hiding property of Commitment Schemes. We give a game-based notion capturing this property which resembles the IND-CPA notion for PKE schemes. For $\mathbf{b} \in \{0,1\}$, $\mathbf{G}_{\mathbf{b}}^{\mathsf{Hiding}}$ provides adversaries with access to oracle $\mathcal{O}_S$ defined above, and to an oracle $\mathcal{O}_{Commit}$ whose behavior is defined below:

Encryption Oracle: $\mathcal{O}_{Commit}(m_0, m_1)$

1. Pick randomness ρ uniformly at random;
2. For game system $\mathbf{G}_{\mathbf{b}}^{\mathsf{Hiding}}$, compute $\texttt{comm} \leftarrow \mathit{Commit}_{\texttt{crs}}(m_{\mathbf{b}}; \rho)$; output $\texttt{comm}$.

Definition 17. *Consider the following game played between an adversary* $\mathbf{A}$ *and a game system* $\mathbf{G}_{\mathbf{b}}^{\mathsf{Hiding}}$*, with* $\mathbf{b} \in \{0,1\}$*:*

– $b' \leftarrow \mathbf{A}^{\mathcal{O}_S, \mathcal{O}_{Commit}}$

$\mathbf{A}$ *wins the game if* $b' = \mathbf{b}$*.*

We define the advantage of $\mathbf{A}$ *in winning the* Hiding *game as*

$$\mathit{Adv}^{\mathsf{Hiding}}(\mathbf{A}) := \left| \Pr[\mathbf{AG}_{\mathbf{0}}^{\mathsf{Hiding}} = \texttt{win}] + \Pr[\mathbf{AG}_{\mathbf{1}}^{\mathsf{Hiding}} = \texttt{win}] - 1 \right|.$$

An adversary $\mathbf{A}$ $(\varepsilon_{\mathsf{Hiding}}, t)$-breaks the (q)-Hiding property of a CS Π if it runs in time t, makes at most q queries to $\mathcal{O}_{Commit}$, and satisfies $\mathit{Adv}^{\mathsf{Hiding}}(\mathbf{A}) \geq \varepsilon_{\mathsf{Hiding}}$.

Definition 18, which captures the binding property of Commitment Schemes, provides adversaries with access to an oracle $\mathcal{O}_{Commit}$ defined as follows:

Commit Oracle: $\mathcal{O}_{Commit}(m, \rho)$

1. Compute $\texttt{comm} = \mathit{Commit}_{\texttt{crs}}(m; \rho)$;[12] output $\texttt{comm}$.

Definition 18. *Consider the following game played between an adversary* $\mathbf{A}$ *and game system* $\mathbf{G}^{\mathsf{Binding}}$*:*

– $\mathbf{A}^{\mathcal{O}_S, \mathcal{O}_{Commit}}$

$\mathbf{A}$ *wins the game if there are two queries* q *and* q' *to* $\mathcal{O}_{Commit}$ *where* q *has input* (m, ρ) *and outputs* $\texttt{comm}$ *and* q' *has input* (m', ρ') *and outputs* $\texttt{comm}'$*, satisfying* $m \neq m'$ *and* $\texttt{comm} = \texttt{comm}'$*.*

The advantage of $\mathbf{A}$ *in winning the Binding game is denoted* $\mathit{Adv}^{\mathsf{Binding}}(\mathbf{A})$ *and corresponds to the probability that* $\mathbf{A}$ *wins game* $\mathbf{G}^{\mathsf{Binding}}$ *as described above.*

An adversary $\mathbf{A}$ $(\varepsilon_{\mathsf{Binding}}, t)$-breaks the Binding property of a CS Π if $\mathbf{A}$ runs in time at most t and satisfies $\mathit{Adv}^{\mathsf{Binding}}(\mathbf{A}) \geq \varepsilon_{\mathsf{Binding}}$. If $\mathbf{A}$ is computationally

[12] Here, ρ denotes the random coins used by *Commit*, meaning that $\mathit{Commit}_{(\cdot)}(\cdot; \rho)$ is a deterministic algorithm.

unbounded, we instead write that $\mathbf{A}$ $(\varepsilon_{\mathsf{Binding}})$-breaks the $\mathsf{Binding}$ property of Π if $Adv^{\mathsf{Binding}}(\mathbf{A}) \geq \varepsilon_{\mathsf{Binding}}$.

We say a CS Π is $(\varepsilon_{\mathsf{Hiding}}, \varepsilon_{\mathsf{Binding}}, t, q)$-secure if no adversary $\mathbf{A}$ $(\varepsilon_{\mathsf{Hiding}}, t)$-breaks the (q)-Hiding property of Π, or $(\varepsilon_{\mathsf{Binding}}, t)$-breaks the $\mathsf{Binding}$ property of Π. For a Statistically Binding Commitment Scheme (i.e. one for which the $\mathsf{Binding}$ property holds against computationally unbounded adversaries), we say instead that Π is $(\varepsilon_{\mathsf{Hiding}}, \varepsilon_{\mathsf{Binding}}, t, q, \mathsf{Binding})$-secure if no adversary $\mathbf{A}$ $(\varepsilon_{\mathsf{Hiding}}, t)$-breaks the (q)-Hiding property of Π, and no (possibly computationally unbounded) adversary $(\varepsilon_{\mathsf{Binding}})$-breaks the $\mathsf{Binding}$ property of Π.

D Game-Based Security Definitions for Non Interactive Zero Knowledge Schemes

For a binary relation R, let L_R be the language $L_R \coloneqq \{x \mid \exists w, (x, w) \in R\}$ induced by R. A *Non Interactive Proof System* (NIPS) for L_R is a triple of PPT algorithms $\Pi = (G_{CRS}, \mathit{Prove}, \mathit{Verify})$ where:

- $G_{CRS}(1^k)$: given security parameter 1^k, outputs a common reference string $\mathtt{crs}$;
- $\mathit{Prove}_{\mathtt{crs}}(x, w)$: given a common reference string $\mathtt{crs}$ and a statement-witness pair $(x, w) \in R$, outputs a proof p;
- $\mathit{Verify}_{\mathtt{crs}}(x, p)$: given a common reference string $\mathtt{crs}$, a statement x and a proof p, either accepts, outputting $\mathtt{valid}$ $(= 1)$ or rejects, outputting $\mathtt{invalid}$ $(= 0)$.

In the following definitions, let $\Pi = (G_{CRS}, \mathit{Prove}, \mathit{Verify})$ be a NIPS for a relation R, and let k be the security parameter. The security notions below (Definitions 19 and 20) provide adversaries with access to oracles $\mathcal{O}_S$ and $\mathcal{O}_V$, defined as:

CRS Generation Oracle: $\mathcal{O}_S$

1. On the first call, compute and store $\mathtt{crs} \leftarrow G_{CRS}(1^k)$; output $\mathtt{crs}$;
2. On subsequent calls, output the previously generated $\mathtt{crs}$.

Verify Oracle: $\mathcal{O}_V(x, p)$

1. Compute $b = \mathit{Verify}_{\mathtt{crs}}(x, p)$; output b.

Definition 19 additionally provides adversaries with access to an oracle $\mathcal{O}_P$:

Prove Oracle: $\mathcal{O}_P(x, w)$

1. Compute $p = \mathit{Prove}_{\mathtt{crs}}(x, w)$; output p.

Definition 19. *Consider the following game played between an adversary* $\mathbf{A}$ *and game system* $\mathbf{G}^{\mathsf{Complete}}$*:*

– $\mathbf{A}^{\mathcal{O}_S,\mathcal{O}_P,\mathcal{O}_V}$

A *wins the game if there are two queries* q_P *and* q_V *to* $\mathcal{O}_P$ *and* $\mathcal{O}_V$*, respectively, where* q_P *has input* (x, w) *and* q_V *has input* (x', p)*, satisfying* $x = x'$*, the input* p *in* q_V *is the output of* q_P*, the output of* q_V *is* `invalid`*, and* $(x, w) \in R$*.*

The advantage of **A** *in winning the Completeness game, denoted* $Adv^{\mathsf{Complete}}(\mathbf{A})$*, corresponds to the probability that* **A** *wins game* $\mathbf{G}^{\mathsf{Complete}}$ *as described above.*

We say that an adversary **A** $(\varepsilon_{\mathsf{Complete}}, t)$-breaks the (q_P, q_V)-Completeness of a NIPS scheme Π if **A** runs in time at most t, makes at most q_P and q_V queries to oracles $\mathcal{O}_P$ and $\mathcal{O}_V$, respectively, and satisfies $Adv^{\mathsf{Complete}}(\mathbf{A}) \geq \varepsilon_{\mathsf{Complete}}$.

Definition 20. *Consider the following game played between an adversary* **A** *and game system* $\mathbf{G}^{\mathsf{Sound}}$*:*

– $\mathbf{A}^{\mathcal{O}_S,\mathcal{O}_V}$

A *wins the game if there is a query to* $\mathcal{O}_V$ *on input* (x, p)*, satisfying* $x \notin L_R$*, such that the oracle outputs* `valid`*.*

The advantage of **A** *in winning the Soundness game corresponds to the probability that* **A** *wins game* $\mathbf{G}^{\mathsf{Sound}}$ *as described above and is denoted* $Adv^{\mathsf{Sound}}(\mathbf{A})$*.*

An adversary **A** $(\varepsilon_{\mathsf{Sound}}, t)$-breaks the (q_V)-Soundness of a NIPS scheme Π if **A** runs in time at most t, makes at most q_V queries to $\mathcal{O}_V$ and satisfies $Adv^{\mathsf{Sound}}(\mathbf{A}) \geq \varepsilon_{\mathsf{Sound}}$.

A NIZK scheme $\Pi = (G_{CRS}, \mathit{Prove}, \mathit{Verify}, S = (S_{CRS}, S_{Sim}))$ for a relation R consists of a NIPS scheme $\Pi' = (G_{CRS}, \mathit{Prove}, \mathit{Verify})$ for R and a simulator $S = (S_{CRS}, S_{Sim})$, where:

– $S_{CRS}(1^k)$: given security parameter 1^k, outputs a pair $(\mathtt{crs}, \tau)$;
– $S_{Sim(\mathtt{crs},\tau)}(x)$: given a pair $(\mathtt{crs}, \tau)$ and a statement x, outputs a proof p.

Consider a NIZK scheme $\Pi = (G_{CRS}, \mathit{Prove}, \mathit{Verify}, S = (S_{CRS}, S_{Sim}))$. The following security notion, which defines game systems $\mathbf{G}_{\mathbf{0}}^{\mathsf{ZK}}$ and $\mathbf{G}_{\mathbf{1}}^{\mathsf{ZK}}$, provides adversaries with access to two oracles, $\mathcal{O}_S$ and $\mathcal{O}_P$, whose behavior depends on the underlying game system. For $\mathbf{G}_{\mathbf{b}}^{\mathsf{ZK}}$ (with $\mathbf{b} \in \{0, 1\}$):

CRS Generation Oracle: $\mathcal{O}_S$

1. On the first call, compute and store $\mathtt{crs} \leftarrow G_{CRS}(1^k)$ if $\mathbf{b} = \mathbf{0}$, and $(\mathtt{crs}, \tau) \leftarrow S_{CRS}(1^k)$ if $\mathbf{b} = \mathbf{1}$; output `crs`;
2. On subsequent calls, output the previously generated `crs`.

Prove Oracle: $\mathcal{O}_P(x, w)$

– If $\mathbf{b} = \mathbf{0}$, output $\pi \leftarrow \mathit{Prove}_{\mathtt{crs}}(x, w)$;
– If $\mathbf{b} = \mathbf{1}$, output $\pi \leftarrow S_{Sim(\mathtt{crs},\tau)}(x)$.

Definition 21. *For* $\mathbf{b} \in \{0, 1\}$*, consider the following game played between an adversary* **A** *and game system* $\mathbf{G}_{\mathbf{b}}^{\mathsf{ZK}}$*:*

– $b' \leftarrow \mathbf{A}^{\mathcal{O}_S, \mathcal{O}_P}$

A *wins the game if* $b' = \mathbf{b}$ *and every query* $\mathcal{O}_P(x, w)$ *satisfies* $(x, w) \in R$.

The advantage of **A** *in winning the Zero-Knowledge security game for* Π *is*

$$Adv^{\mathsf{ZK}}(\mathbf{A}) := \Big|\Pr[\mathbf{AG}_{\mathbf{0}}^{\mathsf{ZK}} = \mathtt{win}] + \Pr[\mathbf{AG}_{\mathbf{1}}^{\mathsf{ZK}} = \mathtt{win}] - 1\Big|.$$

We say that an adversary **A** $(\varepsilon_{\mathsf{ZK}}, t)$-breaks the (q_P)-ZK security of a NIZK scheme Π if it makes at most q_P queries to $\mathcal{O}_P$ and satisfies $Adv^{\mathsf{ZK}}(\mathbf{A}) \geq \varepsilon_{\mathsf{ZK}}$.

We now introduce Simulation Soundness for NIZK [37]. The game system defined by this notion provides adversaries with access to oracles $\mathcal{O}_S$, $\mathcal{O}_P$ and $\mathcal{O}_V$ defined as:

CRS Generation Oracle: $\mathcal{O}_S$

1. On the first call, compute and store $(\mathtt{crs}, \tau) \leftarrow S_{CRS}(1^k)$; output $\mathtt{crs}$;
2. On subsequent calls, output the previously generated $\mathtt{crs}$.

Prove Oracle: $\mathcal{O}_P(x)$

1. Compute $p = S_{Sim(\mathtt{crs}, \tau)}(x)$; output p.

Verify Oracle: $\mathcal{O}_V(x, p)$

1. Compute $b = \mathit{Verify}_{\mathtt{crs}}(x, p)$; output b.

Definition 22. *Consider the following game played between an adversary* **A** *and game system* $\mathbf{G}^{\mathsf{SS}}$*:*

– $\mathbf{A}^{\mathcal{O}_S, \mathcal{O}_P, \mathcal{O}_V}$

A *wins the game if it makes a query* $\mathcal{O}_V(x, p)$ *with* $x \notin L_R$ *that outputs* $\mathtt{valid}$ *and no query* $\mathcal{O}_P(x)$ *output* p.

The advantage of **A** *in winning the Simulation Soundness game, denoted* $Adv^{\mathsf{SS}}(\mathbf{A})$*, is the probability that* **A** *wins game* $\mathbf{G}^{\mathsf{SS}}$ *as described above.*

An adversary **A** $(\varepsilon_{\mathsf{SS}}, t)$-breaks the (q_P, q_V)-Simulation Soundness of a NIZK scheme Π if it makes at most q_P and q_V queries to $\mathcal{O}_P$ and $\mathcal{O}_V$, respectively, and satisfies $Adv^{\mathsf{SS}}(\mathbf{A}) \geq \varepsilon_{\mathsf{SS}}$.

Finally, we say that a NIZK scheme Π is $(\varepsilon_{\mathsf{Complete}}, \varepsilon_{\mathsf{Sound}}, \varepsilon_{\mathsf{ZK}}, \varepsilon_{\mathsf{SS}}, t, q_P, q_V)$-secure if no adversary **A** $(\varepsilon_{\mathsf{Complete}}, t)$-breaks the (q_P, q_V)-Completeness of Π, $(\varepsilon_{\mathsf{Sound}}, t)$-breaks the (q_V)-Soundness of Π, $(\varepsilon_{\mathsf{ZK}}, t)$-breaks the (q_P)-Zero-Knowledge of Π, or $(\varepsilon_{\mathsf{SS}}, t)$-breaks the (q_P, q_V)-Simulation Soundness of Π.

E Game-Based Security Definitions for Multi-Designated Verifier Signature Schemes

A Multi-Designated Verifier Signature scheme (MDVS) Π is a 6-tuple $\Pi = (S, G_S, G_V, \mathit{Sig}, \mathit{Vfy}, \mathit{Forge})$. The security games for MDVS schemes have an implicitly defined security parameter k, and provide adversaries with access to some of the following oracles:

Public Parameter Generation Oracle: $\mathcal{O}_{PP}$

1. On the first call to $\mathcal{O}_{PP}$, compute $\mathtt{pp} \leftarrow S(1^k)$; output $\mathtt{pp}$;
2. On subsequent calls, simply output $\mathtt{pp}$.

Signer Key-Pair Generation Oracle: $\mathcal{O}_{SK}(A_i)$

1. On the first call to $\mathcal{O}_{SK}$ on input A_i, compute $(\mathtt{spk}_i, \mathtt{ssk}_i) \leftarrow G_S(\mathtt{pp})$, and output $(\mathtt{spk}_i, \mathtt{ssk}_i)$;
2. On subsequent calls, simply output $(\mathtt{spk}_i, \mathtt{ssk}_i)$.

Verifier Key-Pair Generation Oracle: $\mathcal{O}_{VK}(B_j)$

1. Analogous to the Signer Key-Pair Generation Oracle.

Signer Public-Key Oracle: $\mathcal{O}_{SPK}(A_i)$

1. $(\mathtt{spk}_i, \mathtt{ssk}_i) \leftarrow \mathcal{O}_{SK}(A_i)$; output $\mathtt{spk}_i$.

Verifier Public-Key Oracle: $\mathcal{O}_{VPK}(B_j)$

1. Analogous to the Signer Public-Key Oracle.

Signing Oracle: $\mathcal{O}_S(A_i, \vec{V}, m)$

1. $(\mathtt{spk}_i, \mathtt{ssk}_i) \leftarrow \mathcal{O}_{SK}(A_i)$;
2. $\vec{v} = (\mathcal{O}_{VPK}(V_1), \ldots, \mathcal{O}_{VPK}(V_{|\vec{V}|}))$;
3. Output $\sigma \leftarrow \mathit{Sig}_{\mathtt{pp}}(\mathtt{ssk}_i, \vec{v}, m)$.

Verification Oracle: $\mathcal{O}_V(A_i, B_j \in \mathrm{Set}(\vec{V}), \vec{V}, m, \sigma)$

1. $\mathtt{spk}_i \leftarrow \mathcal{O}_{SPK}(A_i)$;
2. $\vec{v} = (\mathcal{O}_{VPK}(V_1), \ldots, \mathcal{O}_{VPK}(V_{|\vec{V}|}))$;
3. $(\mathtt{vpk}_j, \mathtt{vsk}_j) \leftarrow \mathcal{O}_{VK}(B_j)$;
4. Output $d \leftarrow \mathit{Vfy}_{\mathtt{pp}}(\mathtt{spk}_i, \mathtt{vsk}_j, \vec{v}, m, \sigma)$, where $d \in \{0, 1\}$.

We now introduce the relevant game-based notions for MDVS schemes. Let $\Pi = (S, G_S, G_V, \mathit{Sig}, \mathit{Vfy}, \mathit{Forge})$ be an MDVS scheme.

Definition 23 (Correctness). *Consider the following game being played between an adversary* $\mathbf{A}$ *and game system* $\mathbf{G}^{\mathsf{Corr}}$*:*

1. $\mathbf{A}^{\mathcal{O}_{PP}, \mathcal{O}_{SK}, \mathcal{O}_{VK}, \mathcal{O}_{SPK}, \mathcal{O}_{VPK}, \mathcal{O}_S, \mathcal{O}_V}$

$\mathbf{A}$ *wins the game if there are two queries* q_S *and* q_V *to* $\mathcal{O}_S$ *and* $\mathcal{O}_V$*, respectively, where* q_S *has input* $(A_i, \vec{V}, m)$ *and* q_V *has input* $(A_i{}', B_j, \vec{V}', m', \sigma)$*, satisfying* $(A_i, \vec{V}, m) = (A_i{}', \vec{V}', m')$*,* $B_j \in \vec{V}$*, the input* σ *in* q_V *is the output of the oracle* $\mathcal{O}_S$ *on query* q_S*, and the output of the oracle* $\mathcal{O}_V$ *on the query* q_V *is* 0*.*

The advantage of $\mathbf{A}$ *in winning the Correctness game is the probability that* $\mathbf{A}$ *wins game* $\mathbf{G}^{\mathsf{Corr}}$ *as described above, and is denoted* $\mathit{Adv}^{\mathsf{Corr}}(\mathbf{A})$*.*

Definition 24 (Consistency). *Consider the following game being played between an adversary* $\mathbf{A}$ *and game* $\mathbf{G}^{\mathsf{Cons}}$*:*

1. $\mathbf{A}^{\mathcal{O}_{PP},\mathcal{O}_{SK},\mathcal{O}_{VK},\mathcal{O}_{SPK},\mathcal{O}_{VPK},\mathcal{O}_S,\mathcal{O}_V}$

$\mathbf{A}$ *wins if it queries* $\mathcal{O}_V$ *on inputs* $(A_i, B_j, \vec{V}, m, \sigma)$ *and* $(A_i{}', B_j{}', \vec{V}', m', \sigma')$ *with* $(A_i, \vec{V}, m, \sigma) = (A_i{}', \vec{V}', m', \sigma')$ *and such that* $\{B_j, B_j{}'\} \subseteq \vec{V}$*, the outputs of the two queries differ, and there is no query* $\mathcal{O}_{VK}(B_j)$ *prior to query* $\mathcal{O}_V(A_i, B_j, \vec{V}, m, \sigma)$*, nor query* $\mathcal{O}_{VK}(B_j{}')$ *prior to query* $\mathcal{O}_V(A_i{}', B_j{}', \vec{V}', m', \sigma')$*.*

The advantage of $\mathbf{A}$ *in winning the Consistency game is the probability that* $\mathbf{A}$ *wins game* $\mathbf{G}^{\mathsf{Cons}}$ *as described above, and is denoted* $Adv^{\mathsf{Cons}}(\mathbf{A})$*.*

Remark 13. As mentioned in [12], the Unforgeability notion below is not equivalent to the one introduced in [16] in that it provides adversaries with access to a signature verification oracle. While the notion we assume is stronger than the one originally given in [16], there exist MDVS schemes satisfying this notion, namely the one from [12], and therefore the claims of the paper remain unaffected.

Definition 25 (Unforgeability). *Consider the following game being played between an adversary* $\mathbf{A}$ *and game system* $\mathbf{G}^{\mathsf{Unforg}}$*:*

1. $\mathbf{A}^{\mathcal{O}_{PP},\mathcal{O}_{SK},\mathcal{O}_{VK},\mathcal{O}_{SPK},\mathcal{O}_{VPK},\mathcal{O}_S,\mathcal{O}_V}$

$\mathbf{A}$ *wins if it makes a query* $\mathcal{O}_V({A_i}^*, {B_j}^*, \vec{V}^*, m^*, \sigma^*)$ *with* ${B_j}^* \in \vec{V}^*$ *that outputs* 1*, for every query* $\mathcal{O}_S(A_i{}', \vec{V}', m')$*,* $({A_i}^*, \vec{V}^*, m^*) \neq (A_i{}', \vec{V}', m')$ *and there is no* $\mathcal{O}_{SK}$ *query on* ${A_i}^*$ *nor* $\mathcal{O}_{VK}$ *query on* ${B_j}^*$*.*

$\mathbf{A}$*'s advantage is denoted* $Adv^{\mathsf{Unforg}}(\mathbf{A})$ *and corresponds to the probability that* $\mathbf{A}$ *wins* $\mathbf{G}^{\mathsf{Unforg}}$*.*

The Off-The-Record security notion defines two game systems, $\mathbf{G_0^{OTR}}$ and $\mathbf{G_1^{OTR}}$, which provide adversaries with access to (modified) oracles $\mathcal{O}_S$ and $\mathcal{O}_V$ described below:

Signing Oracle: $\mathcal{O}_S(\mathtt{type} \in \{\mathtt{sig}, \mathtt{sim}\}, A_i, \vec{V}, m, \mathcal{C} \subseteq \mathrm{Set}(\vec{V}))$

For game system $\mathbf{G_b^{OTR}}$, the oracle behaves as follows:

1. $(\mathtt{spk}_i, \mathtt{ssk}_i) \leftarrow \mathcal{O}_{SK}(A_i)$;
2. Let $\vec{v} = (v_1, \ldots, v_{|\vec{V}|})$ and $\vec{s} = (s_1, \ldots, s_{|\vec{V}|})$ where, for $i \in \{1, \ldots, |\vec{V}|\}$:
   - $(v_i, s_i) = \begin{cases} \mathcal{O}_{VK}(V_i) & \text{if } V_i \in \mathcal{C} \\ (\mathcal{O}_{VPK}(V_i), \perp) & \text{otherwise;} \end{cases}$
3. $(\sigma_{\mathbf{0}}, \sigma_{\mathbf{1}}) \leftarrow (\Pi.Sig_{\mathtt{pp}}(\mathtt{ssk}_i, \vec{v}, m), \Pi.Forge_{\mathtt{pp}}(\mathtt{spk}_i, \vec{v}, m, \vec{s}))$;
4. If $\mathbf{b} = 0$, output $\sigma_{\mathbf{0}}$ if $\mathtt{type} = \mathtt{sig}$ and $\sigma_{\mathbf{1}}$ if $\mathtt{type} = \mathtt{sim}$; otherwise, if $\mathbf{b} = 1$, output $\sigma_{\mathbf{1}}$.

Verification Oracle: $\mathcal{O}_V(A_i, B_j \in \mathrm{Set}(\vec{V}), \vec{V}, m, \sigma)$

1. If σ was output by a query to $\mathcal{O}_S$ on an input $(\texttt{type}, A_i', \vec{V}', m', \mathcal{C})$ such that $(A_i', \vec{V}', m') = (A_i, \vec{V}, m)$ and with $B_j \in \vec{V}$, output $\texttt{test}$;
2. Otherwise, compute $b \leftarrow \mathit{Vfy}_{\mathsf{pp}}(\texttt{spk}_i, \texttt{vsk}_j, \vec{v}, m, \sigma)$; output b.

Definition 26 (Off-The-Record). *For $\mathbf{b} \in \{0,1\}$, consider the following game being played between an adversary $\mathbf{A}$ and game system $\mathbf{G}_{\mathbf{b}}^{\mathsf{OTR}}$:*

1. $\mathbf{A}^{\mathcal{O}_{PP},\mathcal{O}_{SK},\mathcal{O}_{VK},\mathcal{O}_{SPK},\mathcal{O}_{VPK},\mathcal{O}_S,\mathcal{O}_V}$

We say that $\mathbf{A}$ wins the game if it outputs a guess bit b' with $b' = \mathbf{b}$, and for every query $\mathcal{O}_S(\texttt{type}, A_i, \vec{V}, m, \mathcal{C})$: 1. there is no query $\mathcal{O}_{VK}(B_j)$ with $B_j \in \mathit{Set}(\vec{V}) \setminus \mathcal{C}$; and 2. there is no query $\mathcal{O}_{SK}(A_i')$ with $A_i' = A_i$.

The advantage of $\mathbf{A}$ in winning the Off-The-Record security game is

$$\mathit{Adv}^{\mathsf{OTR}}(\mathbf{A}) \coloneqq \left|\Pr[\mathbf{AG}_{\mathbf{0}}^{\mathsf{OTR}} = \texttt{win}] + \Pr[\mathbf{AG}_{\mathbf{1}}^{\mathsf{OTR}} = \texttt{win}] - 1\right|.$$

We say an adversary $\mathbf{A}$ (ε, t)-breaks the $(n_S, n_V, d_S, q_S, q_V)$-Correctness (resp. -Consistency, -Unforgeability and -Off-The-Record) of Π if $\mathbf{A}$ runs in time at most t, queries $\mathcal{O}_{SK}$, $\mathcal{O}_{SPK}$, $\mathcal{O}_S$ and $\mathcal{O}_V$ on at most n_S different signers, $\mathcal{O}_{VK}$, $\mathcal{O}_{VPK}$, $\mathcal{O}_S$ and $\mathcal{O}_V$ on at most n_V different verifiers, makes at most q_S and q_V queries to $\mathcal{O}_S$ and $\mathcal{O}_V$, respectively, with the sum of the verifier vectors' lengths input to $\mathcal{O}_S$ being at most d_S, and satisfies $\mathit{Adv}^{\mathsf{Corr}}(\mathbf{A}) \geq \varepsilon$ (resp. $\mathit{Adv}^{\mathsf{Cons}}(\mathbf{A}) \geq \varepsilon$, $\mathit{Adv}^{\mathsf{Unforg}}(\mathbf{A}) \geq \varepsilon$ and $\mathit{Adv}^{\mathsf{OTR}}(\mathbf{A}) \geq \varepsilon$). We say that Π is

$$(\varepsilon_{\mathsf{Corr}}, \varepsilon_{\mathsf{Cons}}, \varepsilon_{\mathsf{Unforg}}, \varepsilon_{\mathsf{OTR}}, t, n_S, n_V, d_S, q_S, q_V)\text{-secure}$$

if no adversary:

- $(\varepsilon_{\mathsf{Corr}}, t)$-breaks the $(n_S, n_V, d_S, q_S, q_V)$-Correctness of Π;
- $(\varepsilon_{\mathsf{Cons}}, t)$-breaks the $(n_S, n_V, d_S, q_S, q_V)$-Consistency of Π;
- $(\varepsilon_{\mathsf{Unforg}}, t)$-breaks the $(n_S, n_V, d_S, q_S, q_V)$-Unforgeability of Π; or
- $(\varepsilon_{\mathsf{OTR}}, t)$-breaks the $(n_S, n_V, d_S, q_S, q_V)$-Off-The-Record security of Π.

E.1 Privacy of Identities

The following security notion is the multi-challenge variant of the *Privacy of Identities* notion from [16]. Similarly to the Off-The-Record security notion, this notion defines two game systems, $\mathbf{G}_{\mathbf{0}}^{\mathsf{PI}}$ and $\mathbf{G}_{\mathbf{1}}^{\mathsf{PI}}$ and provides adversaries with access to (modified) oracles $\mathcal{O}_S$ and $\mathcal{O}_V$ defined below:

Signing Oracle: $\mathcal{O}_S((A_{i,\mathbf{0}}, \vec{V_{\mathbf{0}}}), (A_{i,\mathbf{1}}, \vec{V_{\mathbf{1}}}), m)$

For game system $\mathbf{G}_{\mathbf{b}}^{\mathsf{PI}}$, the oracle behaves as follows:

1. $(\texttt{spk}_{i,\mathbf{b}}, \texttt{ssk}_{i,\mathbf{b}}) \leftarrow \mathcal{O}_{SK}(A_{i,\mathbf{b}})$;
2. $\vec{v_{\mathbf{b}}} = (\mathcal{O}_{VPK}(V_{\mathbf{b},1}), \ldots, \mathcal{O}_{VPK}(V_{\mathbf{b},|\vec{V}|}))$;
3. $\sigma \leftarrow \mathit{Sig}_{\mathsf{pp}}(\texttt{ssk}_{i,\mathbf{b}}, \vec{v_{\mathbf{b}}}, m)$; output σ.

Verification Oracle: $\mathcal{O}_V(A_i, B_j \in \mathrm{Set}(\vec{V}), \vec{V}, m, \sigma)$

1. If σ was output by a query to $\mathcal{O}_S$, output $\mathtt{test}$;
2. $\mathtt{spk}_i \leftarrow \mathcal{O}_{SPK}(A_i)$;
3. $\vec{v} = (\mathcal{O}_{VPK}(V_1), \ldots, \mathcal{O}_{VPK}(V_{|\vec{V}|}))$;
4. $(\mathtt{vpk}_j, \mathtt{vsk}_j) \leftarrow \mathcal{O}_{VK}(B_j)$;
5. $b \leftarrow \mathit{Vfy}_{\mathsf{pp}}(\mathtt{spk}_i, \mathtt{vsk}_j, \vec{v}, m, \sigma)$; output b.

Definition 27 (Privacy of Identities). *For $\mathbf{b} \in \{0,1\}$, consider the following game played between an adversary $\mathbf{A}$ and $\mathbf{G}^{\mathsf{PI}}_{\mathbf{b}}$:*

1. $\mathbf{A}^{\mathcal{O}_{PP},\mathcal{O}_{SK},\mathcal{O}_{VK},\mathcal{O}_{SPK},\mathcal{O}_{VPK},\mathcal{O}_S,\mathcal{O}_V}$

$\mathbf{A}$ wins the game if it outputs a guess bit b' with $b' = \mathbf{b}$ and for every query $((A_{i,\mathbf{0}}, \vec{V_\mathbf{0}}), (A_{i,\mathbf{1}}, \vec{V_\mathbf{1}}), m)$ to oracle $\mathcal{O}_S$: 1. $|\vec{V_\mathbf{0}}| = |\vec{V_\mathbf{1}}|$; 2. for all queries B_j to $\mathcal{O}_{VK}$, $B_j \notin Set(\vec{V_\mathbf{0}}) \cup Set(\vec{V_\mathbf{1}})$; and 3. for all queries $A_i{}'$ to $\mathcal{O}_{SK}$, $A_i{}' \notin \{A_{i,\mathbf{0}}, A_{i,\mathbf{1}}\}$.

We define the advantage of $\mathbf{A}$ as

$$\mathit{Adv}^{\mathsf{PI}}(\mathbf{A}) \coloneqq \left|\Pr[\mathbf{AG}^{\mathsf{PI}}_{\mathbf{0}} = \mathtt{win}] + \Pr[\mathbf{AG}^{\mathsf{PI}}_{\mathbf{1}} = \mathtt{win}] - 1\right|.$$

An adversary $\mathbf{A}$ (ε, t)-breaks the $(n_S, n_V, d_S, q_S, q_V)$-Identity-Privacy of Π if $\mathbf{A}$ runs in time at most t, queries $\mathcal{O}_{SK}$, $\mathcal{O}_{SPK}$, $\mathcal{O}_S$ and $\mathcal{O}_V$ on at most n_S different signers, $\mathcal{O}_{VK}$, $\mathcal{O}_{VPK}$, $\mathcal{O}_S$ and $\mathcal{O}_V$ on at most n_V different verifiers, makes at most q_S and q_V queries to $\mathcal{O}_S$ and $\mathcal{O}_V$, respectively, with the sum of the verifier vectors' lengths input to $\mathcal{O}_S$ being at most d_S and the advantage of $\mathbf{A}$ in winning the game being at least ε (i.e. $\mathit{Adv}^{\mathsf{PI}}(\mathbf{A}) \geq \varepsilon$). Finally, we say that Π is

$$(\varepsilon_{\mathsf{Corr}}, \varepsilon_{\mathsf{Cons}}, \varepsilon_{\mathsf{Unforg}}, \varepsilon_{\mathsf{OTR}}, \varepsilon_{\mathsf{PI}}, t, n_S, n_V, d_S, q_S, q_V)\text{-secure}$$

if it is

$$(\varepsilon_{\mathsf{Corr}}, \varepsilon_{\mathsf{Cons}}, \varepsilon_{\mathsf{Unforg}}, \varepsilon_{\mathsf{OTR}}, t, n_S, n_V, d_S, q_S, q_V)\text{-secure}$$

and no adversary $\mathbf{A}$ $(\varepsilon_{\mathsf{PI}}, t)$-breaks the $(n_S, n_V, d_S, q_S, q_V)$-Privacy of Π.

F Multi-Designated Verifier Signature Scheme with Privacy from Standard Assumptions

The construction of an MDVS scheme achieving Privacy of Identities (see Definition 27) from standard assumptions is straightforward from the MDRS-PKE scheme construction given in Algorithm 2.[13] For completeness, we give an explicit construction in Algorithm 3.

[13] In particular, one can reduce any adversary breaking the privacy of identities of the MDVS to one breaking the IK-CCA-2 security of the underlying MDRS-PKE.

Algorithm 3 MDVS construction $\Pi = (S, G_S, G_V, Sig, Vfy, Forge)$ from MDRS-PKE scheme $\Pi_{\text{MDRS-PKE}} = (S, G_S, G_R, E, D, Forge)$.

$S(1^k)$
  return $\Pi_{\text{MDRS-PKE}}.S(1^k)$

$G_S(\mathtt{pp})$
  return $\Pi_{\text{MDRS-PKE}}.G_S(\mathtt{pp})$

$G_V(\mathtt{pp})$
  return $\Pi_{\text{MDRS-PKE}}.G_R(\mathtt{pp})$

$Sig(\mathtt{pp}, \mathtt{ssk}_i, \vec{v}, m)$
  return $\Pi_{\text{MDRS-PKE}}.E_{\mathtt{pp}}(\mathtt{ssk}_i, \vec{v}, m)$

$Vfy(\mathtt{pp}, \mathtt{spk}_i, \mathtt{rsk}_j, \vec{v}, m, c)$
  return $(\mathtt{spk}_i, \vec{v}, m) \stackrel{?}{=} \Pi_{\text{MDRS-PKE}}.D_{\mathtt{pp}}(\mathtt{rsk}_j, c)$

$Forge_{\mathtt{pp}}(\mathtt{spk}_i, \vec{v}, m, \vec{s})$
  return $\Pi_{\text{MDRS-PKE}}.Forge_{\mathtt{pp}}(\mathtt{spk}_i, \vec{v}, m, \vec{s})$

G Tight Multi-User Multi-Challenge IK-CPA Security of ElGamal

Throughout this section, let $G = \langle g \rangle$ be some fixed cyclic group of prime order q (i.e. $|G| = q$), and g be a (fixed) generator of G.

Definition 28. *For* $\mathbf{b} \in \{0, 1\}$*, consider the following game between an adversary* $\mathbf{A}$ *and* $\mathbf{G}_{\mathbf{b}}^{\mathsf{DDH}}$*:*

1. $(x, y, z) \xleftarrow{\$} \mathbb{Z}_q \times \mathbb{Z}_q \times \mathbb{Z}_q$*;*
2. $(X, Y, Z_0, Z_1) = (g^x, g^y, g^{xy}, g^z)$*;*
3. $b' \leftarrow \mathbf{A}(X, Y, Z_{\mathbf{b}})$*.*

$\mathbf{A}$ *wins the game if* $b' = \mathbf{b}$*.* $\mathbf{A}$*'s advantage in winning* DDH *is:*

$$Adv^{\mathsf{DDH}}(\mathbf{A}) := \Big|\Pr[\mathbf{AG}_{\mathbf{0}}^{\mathsf{DDH}} = \mathtt{win}] + \Pr[\mathbf{AG}_{\mathbf{1}}^{\mathsf{DDH}} = \mathtt{win}] - 1\Big|.$$

We say that an adversary $\mathbf{A}$ $(\varepsilon_{\mathsf{DDH}}, t)$-breaks the DDH assumption (for group G) if $\mathbf{A}$ runs in time at most t and satisfies $Adv^{\mathsf{DDH}}(\mathbf{A}) \geq \varepsilon_{\mathsf{DDH}}$; conversely we say that the DDH assumption $(\varepsilon_{\mathsf{DDH}}, t)$-holds (for G) if no adversary $\mathbf{A}$ $(\varepsilon_{\mathsf{DDH}}, t)$-breaks the DDH assumption (for G).

For the cyclic group G from above, the ElGamal [18] PKE scheme works as follows:

Key-Pair Generation

1. Pick $b \leftarrow \mathbb{Z}_q$ uniformly at random, and compute $B = g^b$;
2. Output the key-pair $(\mathtt{pk} := B, \mathtt{sk} := b)$.

Encryption

1. On input ($\mathtt{pk} \coloneqq B, m \in G$), pick $r \leftarrow \mathbb{Z}_q$ uniformly at random;
2. Output as ciphertext $c \coloneqq (\alpha \coloneqq g^r, \beta \coloneqq B^r \cdot m)$.

Decryption

1. On input ($\mathtt{sk} \coloneqq b, c \coloneqq (\alpha, \beta) \in G^2$), compute α^{-b};
2. Output $\alpha^{-b} \cdot \beta$.

It is well known that ElGamal is tightly Multi-User Multi-Challenge IND-CPA secure under DDH [8]: if an adversary $\mathbf{A}$ (ε, t)-breaks the (n, q_E)-IND-CPA security of ElGamal, then there is an adversary $\mathbf{A}'$ that (ε', t')-breaks the DDH assumption (for the same group G) with $\varepsilon' \geq \varepsilon/2 - (n+1)/q$ and $t' = t + O(T^{\text{exp}} \cdot (n + q_E))$, [14] where T^{exp} is an upper bound on the time to compute an exponentiation.[15] However, to the best of our knowledge only the Single-User Single-Challenge IK-CPA security of ElGamal has been proven [7]. For completeness, we show that ElGamal is also tightly Multi-User Multi-Challenge IK-CPA secure under DDH. Our proof will rely on the self-reducibility of DDH, and in particular we will use the following helpful lemma from [8, Lemma 5.2] (adapted to the fixed group G from above):

Lemma 1 (Adapted from [8, Lemma 5.2]). *There is a probabilistic algorithm R running in time $O(T^{\text{exp}})$—where T^{exp} is an upper bound on the time to compute an exponentiation in G—that, on input $(g^a, g^b, g^c, x) \in G^3 \times \{0,1\}$, where $a, b, c \in \mathbb{Z}_q$, outputs a triple $(g^{a'}, g^{b'}, g^{c'}) \in G^3$, where $a', b', c' \in \mathbb{Z}_q$, satisfying:*

<table>
<tr><th></th><th>$c = a \cdot b \bmod q$</th><th>$c \neq a \cdot b \bmod q$</th></tr>
<tr><td rowspan="2">$x = 0$</td><td colspan="2">$a' = a$
b' is random</td></tr>
<tr><td>$c' = a' \cdot b' \bmod q$</td><td>c' is random</td></tr>
<tr><td rowspan="2">$x = 1$</td><td colspan="2">a' is random
b' is random</td></tr>
<tr><td>$c' = a' \cdot b' \bmod q$</td><td>c' is random</td></tr>
</table>

In the table, random means that the element is distributed uniformly over $\mathbb{Z}_q$, independently of anything else.

Theorem 12. *If there is an adversary $\mathbf{A}$ that (ε, t)-breaks the (n, q_E)-IK-CPA security of the ElGamal* PKE *scheme, then there is an adversary $\mathbf{A}'$ that (ε', t')-breaks the* DDH *assumption (for the same group G), with*

[14] In [8] the adversary's runtime is actually bounded by $t' = t + O(T^{\text{exp}} \cdot (n \cdot q_E))$. However, by a simple adaptation of their reduction (analogous to the one we give in the proof of Theorem 12) one can actually obtain bound $t' = t + O(T^{\text{exp}} \cdot (n + q_E))$.

[15] The advantage bounds stated in [8, Theorem 5.3] are slightly different. We wrote different bounds here because we found a *minor* issue with the security proof (that leads to slightly different bounds, see Remark 14). Nevertheless the claims made in [8] remain unaffected.

– $\varepsilon' \geq \varepsilon/2 - {}^{(n+1)}/_{q}$; *and*

– $t' = t + O(T^{\text{exp}} \cdot (n + q_E))$, *where* T^{exp} *is an upper bound on the time to compute an exponentiation.*

Proof. Consider the IK-CPA game systems $\mathbf{G}_{\mathbf{0}}^{\mathsf{IK\text{-}CPA}}$ and $\mathbf{G}_{\mathbf{1}}^{\mathsf{IK\text{-}CPA}}$ for the ElGamal PKE scheme. We give reductions $\mathbf{C}_0$ and $\mathbf{C}_1$ satisfying:

(1) $\mathbf{C}_0\mathbf{G}_{\mathbf{1}}^{\mathsf{DDH}} \equiv \mathbf{C}_1\mathbf{G}_{\mathbf{1}}^{\mathsf{DDH}}$ with probability at least $1 - {}^{(n+1)}/_{q}$;

(2) $\mathbf{C}_0\mathbf{G}_{\mathbf{0}}^{\mathsf{DDH}} \equiv \mathbf{G}_{\mathbf{0}}^{\mathsf{IK\text{-}CPA}}$;

(3) $\mathbf{C}_1\mathbf{G}_{\mathbf{0}}^{\mathsf{DDH}} \equiv \mathbf{G}_{\mathbf{1}}^{\mathsf{IK\text{-}CPA}}$.

We now specify reductions $\mathbf{C}_0$ and $\mathbf{C}_1$. In the following, let R be the probabilistic algorithm from Lemma 1, and (X, Y, Z) be the DDH challenge given by $\mathbf{G}^{\mathsf{DDH}}$:

– $\mathcal{O}_{PK}(B_j)$:

1. On the first call on B_j, run R on $(X, Y, Z, 1)$, to obtain $(X_{B_j}, Y_{B_j}, Z_{B_j})$; store $(B_j, (X_{B_j}, Y_{B_j}, Z_{B_j}))$; output $\mathtt{pk}_j \coloneqq X_{B_j}$;
2. On subsequent calls for B_j, simply output $\mathtt{pk}_j \coloneqq X_{B_j}$.

– $\mathcal{O}_E(B_{j,0}, B_{j,1}, m)$:
For $b \in \{0, 1\}$, reduction $\mathbf{C}_b$ proceeds as follows:

1. For $b' \in \{0, 1\}$, let $(B_{j,b'}, (X_{B_{j,b'}}, Y_{B_{j,b'}}, Z_{B_{j,b'}}))$ be the tuple stored by $\mathcal{O}_{PK}$ for party $B_{j,b'}$ (if there is no such tuple, issue query $\mathcal{O}_{PK}(B_{j,b'})$);
2. Run R on input $(X_{B_{j,b'}}, Y_{B_{j,b'}}, Z_{B_{j,b'}}, 0)$, obtaining a triple (X', Y', Z'); output $c \coloneqq (\alpha, \beta)$, where $\alpha = Y'$ and $\beta = Z' \cdot m$.

– When $\mathbf{A}$ outputs a guess $b_{\mathbf{A}}$:

1. $\mathbf{C}_b$ outputs $b_{\mathbf{A}} \oplus b$ as the guess.

It is easy to see, by taking into account the table defined in Lemma 1, that $\mathbf{C}_0$ and $\mathbf{C}_1$ satisfy Conditions (2) and (3) specified above:

– parties' public keys are independent from each other and are distributed uniformly at random over G;

– each ciphertext $c \coloneqq (\alpha, \beta)$ output by $\mathcal{O}_E$ when queried on input $(B_{j,0}, B_{j,1}, m)$ is such that 1. α is independent from everything else and is distributed uniformly at random over G; and 2. if $\log_g(X) = \log_Y(Z)$ then, for $b' \in \{0, 1\}$, letting (X', Y', Z') be the triple generated by reduction $\mathbf{C}_{b'}$ using algorithm R, where X' is $B_{j,b'}$'s public key and $Y' \coloneqq g^a$ is α, we have $\beta = Z' \cdot m = X'^a \cdot m$, implying it has exactly the same distribution as $\mathbf{G}_{\mathbf{b'}}^{\mathsf{IK\text{-}CPA}}$.

At this point, it only remains to prove that condition (1) is satisfied. To that end we will define an event SAME-DIST such that if it occurs condition (1) is satisfied, and will then bound its probability.

Let NR be the event that $\log_g(X) = \log_Y(Z)$. For each party B_j that an adversary queries, reductions $\mathbf{C}_0$ and $\mathbf{C}_1$ sample a triple $(X_{B_j}, Y_{B_j}, Z_{B_j})$ by running R on input $(X, Y, Z, 1)$, and set X_{B_j} as B_j's public key: let $P_1, \ldots, P_n$ be the n distinct parties that the adversary queried for, and for $i = 1, \ldots, n$, let (X'_i, Y'_i, Z'_i) be the triple sampled by the reduction for party P_i (meaning that X'_i is P_i's public key). We define NR_i as the event that $\log_g(X'_i) = \log_{Y'_i}(Z'_i)$. Finally, we define event $\mathsf{SAME\text{-}DIST}$:

$$\mathsf{SAME\text{-}DIST} \coloneqq \overline{\mathsf{NR}} \wedge \left(\bigwedge_{i \in \{1,\ldots,n\}} \overline{\mathsf{NR}_i} \right).$$

It is easy to see that if $\mathsf{SAME\text{-}DIST}$ occurs all the adversary sees is independent of which reduction ($\mathbf{C}_0$ or $\mathbf{C}_1$) it is interacting with—note in particular that the second group element β of each ciphertext $c = (\alpha, \beta)$ is distributed uniformly at random over G—and so indeed condition (1) is satisfied.

We now obtain a lower bound on $\Pr[\mathsf{SAME\text{-}DIST}]$. Consider the opposite event, $\overline{\mathsf{SAME\text{-}DIST}}$. We have

$$\begin{aligned}
\Pr[\overline{\mathsf{SAME\text{-}DIST}}] &= \Pr\left[\mathsf{NR} \vee \left(\bigvee_{i \in \{1,\ldots,n\}} \mathsf{NR}_i \right)\right] \\
&= \Pr[\mathsf{NR}] + \Pr\left[\bigvee_{i \in \{1,\ldots,n\}} \mathsf{NR}_i \;\middle|\; \overline{\mathsf{NR}} \right] \\
&\leq \Pr[\mathsf{NR}] + \sum_{i \in \{1,\ldots,n\}} \Pr[\mathsf{NR}_i \mid \overline{\mathsf{NR}}]
\end{aligned}$$

where the last step follows from the union bound. To conclude the proof, note that since the triple (X, Y, Z) output by $\mathbf{G}_1^{\mathsf{DDH}}$ is such that X, Y and Z are all picked uniformly at random (from G) and independently from each other from G, we have $\Pr[\mathsf{NR}] = 1/q$, and that from Lemma 1 we have $\forall i \in \{1, \ldots, n\}$, $\Pr[\mathsf{NR}_i \mid \overline{\mathsf{NR}}] = 1/q$; so, with probability at least $1 - (n+1)/q$, $\mathbf{C}_0\mathbf{G}_1^{\mathsf{DDH}} \equiv \mathbf{C}_1\mathbf{G}_1^{\mathsf{DDH}}$. □

Remark 14. We noticed a minor issue with the exact bounds claimed in [8, Theorem 5.3] regarding the tight $\mathsf{IND\text{-}CPA}$ security of ElGamal under DDH. In the paper it is claimed that the adversary $\mathbf{A}'$ has an advantage $\varepsilon' \geq \varepsilon/2 - 1/2^k$, where $q \leq 2^k$ (q being the order of group G). However, in the proof of [8, Theorem 5.3], when making an argument analogous to ours establishing that $\mathbf{C}_0\mathbf{G}_1^{\mathsf{DDH}} \equiv \mathbf{C}_1\mathbf{G}_1^{\mathsf{DDH}}$, it is not considered the possibility that the algorithm R mentioned above, which is used to rerandomize DDH challenges, outputs a triple (X', Y', Z') such that $\log_g(X') = \log_{Y'}(Z')$ when given an input $(X, Y, Z, \cdot)$ such that $\log_g(X) \neq \log_Y(Z)$. By Lemma 1, if $\log_g(X) \neq \log_Y(Z)$ then the triple (X', Y', Z') output by R will be one where Z' is uniform at random over G and is independent of all X, Y, Z, X' and Y'; the event mentioned above actually occurs with probability $1/q$. Suppose the event occurs when reduction $\mathbf{C}_b$ is sampling the

public key of a party B_i, and let (X_i, Y_i, Z_i) be the corresponding DDH triple—i.e. (X_i, Y_i, Z_i) was output by R on input $(X, Y, Z, 1)$, with $\log_g(X) \neq \log_Y(Z)$, where X_i is B_i's public key and (X_i, Y_i, Z_i) satisfies $\log_g(X_i) = \log_{Y_i}(Z_i)$: for each query the adversary makes to $\mathcal{O}_E$ on input $(B_{j,0}, B_{j,1}, m)$ with $B_{j,b} = B_i$, by Lemma 1, the DDH triple (X', Y', Z') sampled by R on input $(X_i, Y_i, Z_i, 0)$ will satisfy $\log_g(X') = \log_{Y'}(Z')$. This means that in such case the challenge ciphertext $c = (Y', Z')$ does not hide in an information theoretical sense the bit b of $\mathbf{C}_b$, contrary to what is claimed in the proof of [8, Theorem 5.3].

Nevertheless, by following an argument very similar to the one we used to prove that ElGamal is tightly IK-CPA secure under DDH, one can also prove it is tightly IND-CPA secure under DDH (with the same parameters). This means that despite this minor issue, the claims made in the paper remain unchanged.

H PKEBC Construction Security Proofs

In this section we give the (missing) full security proofs for the PKEBC construction given in Sect. 4.

H.1 Helper Claim

Definition 29 (n-Party ε-Public Key Collision-Resistance). PKE *scheme* $\Pi_{\mathrm{PKE}} = (G, E, D)$ *is* n*-Party* ε*-Public Key Collision-Resistant if*

$$\Pr\left[\left|\{\mathtt{pk}_1, \ldots, \mathtt{pk}_n\}\right| < n \;\middle|\; \begin{array}{c}(\mathtt{pk}_1, \mathtt{sk}_1) \leftarrow \Pi_{\mathrm{PKE}}.G(1^k) \\ \ldots \\ (\mathtt{pk}_n, \mathtt{sk}_n) \leftarrow \Pi_{\mathrm{PKE}}.G(1^k)\end{array}\right] \leq \varepsilon.$$

Lemma 2 (Adapted from [13, Lemma 2]). *If* Π_{PKE} *is*

$$(\varepsilon_{\text{PKE-IND-CPA}}, \varepsilon_{\text{PKE-IK-CPA}}, t_{\mathrm{PKE}}, n_{\mathrm{PKE}}, q_{E_{\mathrm{PKE}}})\text{-}secure,$$

with $t_{\mathrm{PKE}} \gtrapprox n_{\mathrm{PKE}} \cdot t_G + t_D$*—where* t_G *and* t_D *are, respectively, the times to run* $\Pi_{\mathrm{PKE}}.G$ *and* $\Pi_{\mathrm{PKE}}.D$*—then* Π_{PKE} *is* n_{PKE}*-Party* ε*-Public Key Collision-Resistant, with* $\varepsilon \leq 2 \cdot \varepsilon_{\text{PKE-IND-CPA}}$.

H.2 Proof of Theorem 1

We prove a stronger result. Namely, we consider an alternative correctness notion for PKEBC schemes that only differs from Definition 1 in that it allows the adversary to query for the secret key of any receiver and still win the game.

This proof proceeds in a sequence of games [9, 38].

$\mathbf{G}^{\mathsf{Corr}} \rightsquigarrow \mathbf{G}^1$**:** The only difference between $\mathbf{G}^{\mathsf{Corr}}$ and $\mathbf{G}^1$ is that in $\mathbf{G}^1$ the $\mathtt{crs}_{\mathrm{CS}}$ output by $\mathcal{O}_S$ is perfectly binding (see Appendix C).

One can reduce distinguishing the two games to breaking the Binding property of the underlying Π_{CS} scheme: since the reduction has access to the secret keys of any party and can pick the randomness for the commitment itself, it can handle any oracle queries. It then follows from Eq. (4.5), that no adversary $(\varepsilon_{\text{CS-Binding}})$-breaks the Binding property of Π_{CS}, implying

$$\left|\Pr[\mathbf{AG}^{\mathsf{Corr}} = \mathtt{win}] - \Pr[\mathbf{AG}^1 = \mathtt{win}]\right| \leq \varepsilon_{\text{CS-Binding}}.$$

$\mathbf{G}^1 \rightsquigarrow \mathbf{G}^2$**:** The only difference between the games is that in $\mathbf{G}^2$, for each ciphertext $c \coloneqq (p, \mathtt{comm}, \vec{c})$ output by $\mathcal{O}_E$ for a query with input $(\vec{V}, m)$, if $\mathcal{O}_D$ is queried on input (B_j, c) with $B_j \in \vec{V}$, $\mathcal{O}_D$ no longer verifies p's validity using $\Pi_{\mathrm{NIZK}}.V$, and instead simply proceeds as if p would verify as being valid.

As before, one can reduce distinguishing the two games to breaking the completeness of the underlying Π_{NIZK} because the reduction has all secret keys and therefore can handle any queries. (Note that for this game hop the reduction does not need to pick the randomness for proving NIZK statements itself.) Since $\mathbf{A}$ can only make up to $q_E \leq q_{P\,\mathrm{NIZK}}$ queries to $\mathcal{O}_E$ and $q_D \leq q_{V\,\mathrm{NIZK}}$ queries to $\mathcal{O}_D$, it follows from Eq. (4.4), that no adversary $(\varepsilon_{\text{NIZK-Complete}}, t_{\mathrm{NIZK}})$-breaks the $(q_{P\,\mathrm{NIZK}}, q_{V\,\mathrm{NIZK}})$-Completeness of Π_{NIZK}, implying

$$\left|\Pr[\mathbf{AG}^1 = \mathtt{win}] - \Pr[\mathbf{AG}^2 = \mathtt{win}]\right| \leq \varepsilon_{\text{NIZK-Complete}}.$$

$\mathbf{G}^2 \rightsquigarrow \mathbf{G}^3$**:** The only difference between the games is that in $\mathbf{G}^3$, when $\mathcal{O}_D$ is queried on an input (B_j, c), where $c \coloneqq (p, \mathtt{comm}, \vec{c})$ was output by a query $\mathcal{O}_E(\vec{V}, m)$ such that $B_j \in \vec{V}$, $\mathcal{O}_D$ no longer tries decrypting each $c_{l,0}$ of $\vec{c}$ satisfying $v_l = \mathtt{pk}_j$ using $\Pi_{\mathrm{PKE}}.D$, and instead simply assumes the output is $(\rho, \vec{v}, m)$, the tuple encrypted by $\Pi_{\mathrm{PKE}}.E$—with ρ being the random coins used by Π_{CS} to compute the commitment $\mathtt{comm}$.

Note that the probability of winning $\mathbf{G}^3$ is 0: consider any query $\mathcal{O}_E(\vec{V}, m)$ and any later query $\mathcal{O}_D(B_j, c)$ where $c = (p, \mathtt{comm}, \vec{c})$ is the output of the first query and where $B_j \in \vec{V}$:

- Since Π_{PKE} is now assumed to be a correct PKE scheme, then for the least $l \in \{1, \ldots, |\vec{c}|\}$ satisfying $V_l = B_j$, B_j's decryption of $c_{l,0}$ of $\vec{c}$ is going to be $(\vec{v}, m)$, where $\vec{v}$ is the vector of public keys corresponding to $\vec{V}$. By the definition of $\Pi.D$ this then implies that if no $(\vec{v}'', m'') \neq (\vec{v}, m)$ is output—corresponding to the decryption of some $c_{l',0}$ where $l' < l$—then $\Pi.D$ outputs $(\vec{v}, m)$;
- Since $\mathtt{crs}_{\mathrm{CS}}$ is binding, for any $(\vec{v}', m') \neq (\vec{v}, m)$ (with $(\vec{v}', m') \neq \perp$) and any ρ':
$$\mathtt{comm} \neq \Pi_{\mathrm{CS}}.\mathit{Commit}_{\mathtt{crs}_{\mathrm{CS}}}(\vec{v}', m'; \rho'),$$
implying $\mathcal{O}_D$ does not output $(\vec{v}', m') \neq (\vec{v}, m)$.

To conclude, it follows from Eq. (4.3) that Π_{PKE} is perfectly correct, implying

$$\Pr[\mathbf{AG}^2 = \mathtt{win}] = \Pr[\mathbf{AG}^3 = \mathtt{win}] = 0.$$

□

H.3 Proof of Theorem 2

We prove a stronger statement. Namely, we consider an alternative Robustness security notion for PKEBC schemes that only differs from Definition 2 in that it now allows the adversary to query for the secret key of any receiver and still win the game.

$\mathbf{G}^{\mathsf{Rob}} \rightsquigarrow \mathbf{G}^1$: The only difference between the games is that in $\mathbf{G}^1$ no two parties have the same public key.

Since $t_{\mathrm{PKE}} \gtrapprox n_{\mathrm{PKE}} \cdot t_G + t_D$ (where t_G and t_D are, respectively, the times to run $\Pi_{\mathrm{PKE}}.G$ and $\Pi_{\mathrm{PKE}}.D$) and $n_{\mathrm{PKE}} \geq 1$, it follows from Eq. (4.6) and Lemma 2 that

$$\left|\Pr[\mathbf{AG}^{\mathsf{Rob}} = \mathtt{win}] - \Pr[\mathbf{AG}^1 = \mathtt{win}]\right| \leq 2 \cdot \varepsilon_{\text{PKE-IND-CPA}}.$$

We now bound $\Pr[\mathbf{AG}^1 = \mathtt{win}]$. First, note that an adversary $\mathbf{A}$ wins this stronger robustness game if there are two queries q_{E} and q_{D} to $\mathcal{O}_E$ and $\mathcal{O}_D$, respectively, where q_{E} has input $(\vec{V}, m)$ and q_{D} has input (B_j, c), satisfying $B_j \notin \vec{V}$, the input c in q_{D} is the output of q_{E}, and the output of q_{D} is $(\vec{v}', m')$ with $(\vec{v}', m') \neq \perp$. By the definition of $\Pi.D$, the output of q_{D} being some pair $(\vec{v}', m') \neq \perp$ implies $\mathtt{pk}_j \in \vec{v}'$; since $B_j \notin \vec{V}$, it follows from the fact that all parties have distinct public keys that $\mathtt{pk}_j \notin \vec{v}$ (where $\mathtt{pk}_j$ is B_j's public key and $\vec{v}$ is the vector of public keys corresponding to the vector of parties $\vec{V}$), implying $(\vec{v}', m') \neq (\vec{v}, m)$. Also by the definition of $\Pi.D$, the output of q_{D} being some pair $(\vec{v}', m') \neq \perp$ implies there is a sequence of random coins ρ' such that

$$\mathtt{comm} = \Pi_{\mathrm{CS}}.Commit_{\mathtt{crs}_{\mathrm{CS}}}(\vec{v}', m'; \rho').$$

Noting that this is only possible if $\mathtt{crs}_{\mathrm{CS}}$ is non-binding, it follows from Eq. (4.7), that no adversary ($\varepsilon_{\text{CS-Binding}}$)-breaks the Binding property of Π_{CS}, implying

$$\Pr[\mathbf{AG}^1 = \mathtt{win}] \leq \varepsilon_{\text{CS-Binding}}.$$

□

H.4 Proof of Theorem 3

We prove a stronger result. Namely, we consider an alternative Consistency security notion for PKEBC schemes that only differs from Definition 3 in that it allows the adversary to query for the secret key of any receiver (and still win the game).

$\mathbf{G}^{\mathsf{Cons}} \rightsquigarrow \mathbf{G}^1$**:** The only difference is that in $\mathbf{G}^1$ the $\mathtt{crs}_{\mathrm{CS}}$ output by $\mathcal{O}_S$ is perfectly binding (see Appendix C).

One can reduce distinguishing the two games to breaking the Binding property of the underlying Π_{CS} scheme because the reduction has access to the secret keys of any party and can pick the randomness for the commitment itself, and so it can handle any oracle queries. It then follows from Eq. (4.10), that no adversary $(\varepsilon_{\text{CS-Binding}})$-breaks the Binding property of Π_{CS}, implying

$$\left|\Pr[\mathbf{AG}^{\mathsf{Cons}} = \mathtt{win}] - \Pr[\mathbf{AG}^1 = \mathtt{win}]\right| \leq \varepsilon_{\text{CS-Binding}}.$$

$\mathbf{G}^1 \rightsquigarrow \mathbf{G}^2$**:** The difference between the games is that in $\mathbf{G}^2$, whenever $\mathcal{O}_D$ is queried on an input (B_j, c), with $c \coloneqq (p, \mathtt{comm}, \vec{c})$, such that $(\mathtt{crs}_{\mathrm{CS}}, \mathtt{comm}, \vec{c}) \notin L_{\mathrm{Cons}}$ ($\mathtt{crs}_{\mathrm{CS}}$ being the one generated by $\mathcal{O}_{PP}$), $\mathcal{O}_D$ outputs $\perp$.

$\mathbf{G}^2$ is perfectly indistinguishable from $\mathbf{G}^1$ unless $\mathbf{A}$ makes a decryption query on a ciphertext $c \coloneqq (p, \mathtt{comm}, \vec{c})$ such that the NIZK proof p verifies as being a valid one for statement $(\mathtt{crs}_{\mathrm{CS}}, \mathtt{comm}, \vec{c}) \in L_{\mathrm{Cons}}$, and with respect to the $\mathtt{crs}_{\mathrm{CS}}$ output by $\mathcal{O}_S$, but $(\mathtt{crs}_{\mathrm{CS}}, \mathtt{comm}, \vec{c}) \notin L_{\mathrm{Cons}}$. As before, one can reduce distinguishing the two games to breaking the soundness of the underlying Π_{NIZK} because the reduction has all secret keys and therefore can handle any queries. Since $\mathbf{A}$ can only make up to $q_D \leq q_{V\,\mathrm{NIZK}}$ queries to $\mathcal{O}_D$, it follows from Eq. (4.9), that no adversary $(\varepsilon_{\text{NIZK-Sound}}, t_{\mathrm{NIZK}})$-breaks the $(q_{V\,\mathrm{NIZK}})$-Soundness of Π_{NIZK}, implying

$$\left|\Pr[\mathbf{AG}^1 = \mathtt{win}] - \Pr[\mathbf{AG}^2 = \mathtt{win}]\right| \leq \varepsilon_{\text{NIZK-Sound}}.$$

To conclude this proof, we will prove the following claim:

Claim. For any adversary $\mathbf{A}$:

$$\Pr[\mathbf{AG}^2 = \mathtt{win}] = 0.$$

Proof. $\mathbf{A}$ wins $\mathbf{G}^2$, if it queries $\mathcal{O}_D$ on inputs (B_i, c) and (B_j, c) for some B_i and B_j (possibly with $B_i = B_j$) and some ciphertext c, and the first query outputs $(\vec{v}, m) \neq \perp$ with $\mathtt{pk}_j \in \vec{v}$ (where $\mathtt{pk}_j$ is B_j's public key), whereas the second outputs either $\perp$ or some $(\vec{v}', m')$ with $(\vec{v}', m') \neq (\vec{v}, m)$.

Consider any two queries $q_{D,i}$ and $q_{D,j}$ that $\mathbf{A}$ makes to $\mathcal{O}_D$ on inputs (B_i, c) and (B_j, c'), respectively, satisfying $c = c'$, and such that $q_{D,i}$ outputs $(\vec{v}_i, m_i)$ with $(\vec{v}_i, m_i) \neq \perp$ and $\mathtt{pk}_j \in \vec{v}_i$. First, note that if $\mathbf{A}$ does not make any two queries satisfying these conditions, then it does not win $\mathbf{G}^2$. In the following, let $c \coloneqq (p, \mathtt{comm}, \vec{c})$ be the ciphertext input to $q_{D,i}$ and $q_{D,j}$.

By the soundness of Π_{NIZK}, there is a vector of public keys $\vec{v}$ and a message m such that $\mathtt{comm}$ is a commitment to $(\vec{v}, m)$, and for every ciphertext $c_{x,b}$ of $\vec{c}$ there is a sequence of random coins $r_{x,b}$ such that $c_{x,b}$ is the Π_{PKE} encryption of $(\rho, \vec{v}, m)$ under key $v_{x,b}$ using $r_{x,b}$ as the encryption's (sequence of) random coins; by the binding of $\mathtt{crs}_{\mathrm{CS}}$, both $\vec{v}$ and m are unique; the definition of $\Pi.D$ implies $(\vec{v}_i, m_i) = (\vec{v}, m)$ and implies the existence of $l, l' \in \{1, \ldots, |\vec{v}|\}$ satisfying, respectively, $\mathtt{pk}_i = v_l$ and $\mathtt{pk}_j = v_{l'}$. Furthermore, by the definition of $\Pi.D$, $q_{D,i}$

(resp. $q_{D,j}$) will not output $(\vec{v}_{i,\alpha}, m_{i,\alpha})$ from $(\rho_{i,\alpha}, \vec{v}_{i,\alpha}, m_{i,\alpha}) \leftarrow \Pi_{\mathrm{PKE}}.D_{\mathtt{sk}_i}(c_{\alpha,0})$ (resp. $(\vec{v}_{j,\beta}, m_{j,\beta})$ from $(\rho_{j,\beta}, \vec{v}_{j,\beta}, m_{j,\beta}) \leftarrow \Pi_{\mathrm{PKE}}.D_{\mathtt{sk}_j}(c_{\beta,0})$) for any α with $v_\alpha \neq \mathtt{pk}_i$ (resp. any β with $v_\beta \neq \mathtt{pk}_j$), because either $\vec{v}_{j,\alpha} \neq \vec{v}$ (resp. $\vec{v}_{j,\beta} \neq \vec{v}$), or $\mathtt{pk}_i \neq v_\alpha$ (resp. $\mathtt{pk}_j \neq v_\beta$). Again from the definition of $\Pi.D$, $q_{D,i}$ outputs, for some $l \in \{1, \ldots, |\vec{v}|\}$, $(\vec{v}_{i,l}, m_{i,l})$ from $(\rho_{i,l}, \vec{v}_{i,l}, m_{i,l}) \leftarrow \Pi_{\mathrm{PKE}}.D_{\mathtt{sk}_i}(c_{l,0})$, with $v_l = \mathtt{pk}_i$ and where $c_{l,0} \in \vec{c}$. Similarly, $q_{D,j}$ either outputs $\perp$, or outputs, for some $l' \in \{1, \ldots, |\vec{v}|\}$, $(\vec{v}_{j,l'}, m_{j,l'})$ from $(\rho_{j,l'}, \vec{v}_{j,l'}, m_{j,l'}) \leftarrow \Pi_{\mathrm{PKE}}.D_{\mathtt{sk}_j}(c_{l',0})$, with $v_{l'} = \mathtt{pk}_j$ and where $c_{l',0} \in \vec{c}$. Note that, since given a fixed sequence of random coins ρ, $\Pi_{\mathrm{CS}}.\mathit{Commit}$ is a deterministic algorithm, if $(\rho_{i,l}, \vec{v}_{i,l}, m_{i,l}) = (\rho_{j,l'}, \vec{v}_{j,l'}, m_{j,l'})$ then the outputs of $q_{D,i}$ and $q_{D,j}$ are the same. Recall from before that the soundness of Π_{NIZK} implies all ciphertexts $c_{x,b}$ of $\vec{c}$ are encryptions of the same triple $(\rho, \vec{v}, m)$ under some sequence of random coins $r_{x,b}$. Since by Eq. (4.8) Π_{PKE} is perfectly correct, $\mathbf{A}$ cannot win $\mathbf{G}^2$.

This concludes the proof of the claim, and thus also of Theorem 3. □

H.5 Proof of Theorem 4

This proof proceeds in a sequence of game hops. For simplicity of notation, we will refer to $\mathbf{G}_\beta^{\mathsf{IK\text{-}CCA\text{-}2}}$ as $\mathbf{G}_\beta^{\mathsf{CCA}}$, for $\beta \in \{0, 1\}$. The winning condition for each of the games we introduce ahead is the same as for $\mathbf{G}_\mathbf{0}^{\mathsf{CCA}}$—in particular, $\mathbf{A}$ has to output 0 to win each of these games. For any given adversary $\mathbf{A}$, we bound $\mathbf{A}$'s advantage

$$\begin{aligned} \mathit{Adv}^{\mathsf{IK\text{-}CCA\text{-}2}}(\mathbf{A}) &\coloneqq \left|\Pr[\mathbf{AG}_\mathbf{0}^{\mathsf{CCA}} = \mathtt{win}] + \Pr[\mathbf{AG}_\mathbf{1}^{\mathsf{CCA}} = \mathtt{win}] - 1\right| \\ &= \left|\Pr[\mathbf{AG}_\mathbf{0}^{\mathsf{CCA}} = \mathtt{win}] - \Pr[\mathbf{AG}_\mathbf{1}^{\mathsf{CCA}} \neq \mathtt{win}]\right| \end{aligned}$$

by bounding: the difference between the probability of $\mathbf{A}$ winning $\mathbf{G}_\mathbf{0}^{\mathsf{CCA}}$ and winning $\mathbf{G}^1$; for $i \in \{1, \ldots, 9\}$, the difference between the probability of $\mathbf{A}$ winning $\mathbf{G}^i$ and winning $\mathbf{G}^{i+1}$; and by bounding the difference between the probability that $\mathbf{A}$ wins $\mathbf{G}^{10}$ and the probability that $\mathbf{A}$ loses $\mathbf{G}_\mathbf{1}^{\mathsf{CCA}}$. In other words, we bound

$$\left|\Pr[\mathbf{AG}_\mathbf{0}^{\mathsf{CCA}} = \mathtt{win}] - \Pr[\mathbf{AG}^1 = \mathtt{win}]\right|,$$

for $i \in \{1, \ldots, 9\}$, bound,

$$\left|\Pr[\mathbf{AG}^i = \mathtt{win}] - \Pr[\mathbf{AG}^{i+1} = \mathtt{win}]\right|,$$

and bound

$$\begin{aligned} &\left|\Pr[\mathbf{AG}^{10} = \mathtt{win}] + (\Pr[\mathbf{AG}_\mathbf{1}^{\mathsf{CCA}} = \mathtt{win}] - 1)\right| \\ = &\left|\Pr[\mathbf{AG}^{10} = \mathtt{win}] - \Pr[\mathbf{AG}_\mathbf{1}^{\mathsf{CCA}} \neq \mathtt{win}]\right|. \end{aligned}$$

Putting things together, this then implies:

$$\begin{aligned} Adv^{\mathsf{IK\text{-}CCA\text{-}2}}(\mathbf{A}) \leq & \left|\Pr[\mathbf{AG}_{\mathbf{0}}^{\mathsf{CCA}} = \mathtt{win}] - \Pr[\mathbf{AG}^{1} = \mathtt{win}]\right| \\ & + \sum_{i=1,\ldots,9} \left|\Pr[\mathbf{AG}^{i} = \mathtt{win}] - \Pr[\mathbf{AG}^{i+1} = \mathtt{win}]\right| \\ & + \left|\Pr[\mathbf{AG}^{10} = \mathtt{win}] - \Pr[\mathbf{AG}_{\mathbf{1}}^{\mathsf{CCA}} \neq \mathtt{win}]\right|. \end{aligned}$$

$\mathbf{G}_{\mathbf{0}}^{\mathsf{CCA}} \rightsquigarrow \mathbf{G}^{1}$**:** This game is just like $\mathbf{G}_{\mathbf{0}}^{\mathsf{CCA}}$, except that $\mathcal{O}_S$ generates $\mathtt{crs}_{\mathrm{NIZK}}$ using S_{CRS}, and for each challenge ciphertext $c^* := (p, \mathtt{comm}, \vec{c})$, the NIZK proof p is generated by S_{Sim}, meaning it is now simulated.

One can reduce an adversary $\mathbf{A}$ distinguishing the two game systems to one breaking the ZK property of the underlying Π_{NIZK}: the reduction has access to all secret keys and therefore can handle any queries the adversary may make; to generate $\mathtt{crs}_{\mathrm{NIZK}}$ and the NIZK proofs of ciphertexts the reduction can rely on the $\mathcal{O}_P$ oracle provided by Π_{NIZK}'s ZK game. Since $\mathbf{A}$ can only make up to $q_E \leq q_{P\,\mathrm{NIZK}}$ queries to $\mathcal{O}_E$, it follows from Eq. (4.12)

$$\left|\Pr[\mathbf{AG}_{\mathbf{0}}^{\mathsf{CCA}} = \mathtt{win}] - \Pr[\mathbf{AG}^{1} = \mathtt{win}]\right| \leq \varepsilon_{\mathrm{NIZK\text{-}ZK}}.$$

$\mathbf{G}^{1} \rightsquigarrow \mathbf{G}^{2}$**:** Game $\mathbf{G}^{2}$ is just like $\mathbf{G}^{1}$, except that for each $l \in \{1, \ldots, |\vec{c}|\}$, ciphertext $c_{l,1}$ of vector $\vec{c} := \big((c_{1,0}, c_{1,1}), \ldots, (c_{|\vec{c}|,0}, c_{|\vec{c}|,1})\big)$ of the challenge ciphertext $c^* := (p, \mathtt{comm}, \vec{c})$ is now an encryption of $(\tilde{\rho}, \vec{v}_1^*, m^*)$—where $\tilde{\rho}$ is some sequence of random coins, *independent of the one* used by $\Pi_{\mathrm{CS}}.\mathit{Commit}$—instead of $(\rho, \vec{v}_0^*, m^*)$—where ρ is *the sequence of random coins used by* $\Pi_{\mathrm{CS}}.\mathit{Commit}$.

We can reduce distinguishing the two game systems (and therefore bound the difference in $\mathbf{A}$'s winning probabilities) to breaking the $\mathsf{IND\text{-}CPA}$ security of the underlying Π_{PKE} scheme: the reduction holds all secret keys and therefore can handle any decryption queries; for generating NIZK proofs the reduction can simply use simulation trapdoor (which does not require knowledge of a witness). Since $d_E \leq q_{E\,\mathrm{PKE}}$ and $n \leq n_{\mathrm{PKE}}$, it follows from Eq. (4.11)

$$\left|\Pr[\mathbf{AG}^{1} = \mathtt{win}] - \Pr[\mathbf{AG}^{2} = \mathtt{win}]\right| \leq \varepsilon_{\mathrm{PKE\text{-}IND\text{-}CPA}}.$$

$\mathbf{G}^{2} \rightsquigarrow \mathbf{G}^{3}$**:** The difference between the games is that in $\mathbf{G}^{3}$, for each $l \in \{1, \ldots, |\vec{c}|\}$, ciphertext $c_{l,1}$ (of vector $\vec{c} := \big((c_{1,0}, c_{1,1}), \ldots, (c_{|\vec{c}|,0}, c_{|\vec{c}|,1})\big)$ of the challenge ciphertext $c^* := (p, \mathtt{comm}, \vec{c})$) is now encrypted under public key $(v_1^*)_{l,1}$, instead of being encrypted under public key $(v_0^*)_{l,1}$.

By an argument similar to the one for $\mathbf{G}^{1} \rightsquigarrow \mathbf{G}^{2}$, one can reduce distinguishing the two games to breaking the $\mathsf{IK\text{-}CPA}$ security of the underlying Π_{PKE}. Since $d_E \leq q_{E\,\mathrm{PKE}}$ and $n \leq n_{\mathrm{PKE}}$, it again follows from Eq. (4.11) that

$$\left|\Pr[\mathbf{AG}^{2} = \mathtt{win}] - \Pr[\mathbf{AG}^{3} = \mathtt{win}]\right| \leq \varepsilon_{\mathrm{PKE\text{-}IK\text{-}CPA}}.$$

$\mathbf{G}^3 \rightsquigarrow \mathbf{G}^4$**:** The difference between the games is that in $\mathbf{G}^4$ decryption queries for a party B_j—with secret key $((\mathtt{pk}_{j,0}, \mathtt{sk}_{j,0}), (\mathtt{pk}_{j,1}, \mathtt{sk}_{j,1}))$—on a ciphertext $c \coloneqq (p, \mathtt{comm}, \vec{c})$—with $\vec{c} \coloneqq \big((c_{1,0}, c_{1,1}), \ldots, (c_{|\vec{c}|,0}, c_{|\vec{c}|,1})\big)$—behave slightly differently: rather than decrypting, for $l \in \{1, \ldots, |\vec{c}|\}$, ciphertext $c_{l,0}$ using $\mathtt{sk}_{j,0}$, it now decrypts $c_{l,1}$ using $\mathtt{sk}_{j,1}$ instead.

It is easy to see that $\mathbf{G}^3$ and $\mathbf{G}^4$ are perfectly indistinguishable unless $\mathbf{A}$ makes a decryption query for a receiver B_j on ciphertext $c \coloneqq (p, \mathtt{comm}, \vec{c})$ such that the NIZK proof p verifies with respect to $(\mathtt{crs}_{\mathrm{CS}}, \mathtt{comm}, \vec{c})$ ($\mathtt{crs}_{\mathrm{CS}}$ being the one generated by $\mathcal{O}_{PP}$), but $(\mathtt{crs}_{\mathrm{CS}}, \mathtt{comm}, \vec{c}) \notin L_{\mathrm{Cons}}$. So, one can reduce distinguishing $\mathbf{G}^3$ and $\mathbf{G}^4$ to breaking the simulation soundness of the underlying Π_{NIZK}: on one hand the reduction holds all secret keys, so it can handle any decryption queries[16]; on the other hand, even though the reduction does not know the simulation trapdoor, it can rely on the $\mathcal{O}_P$ oracle provided by Π_{NIZK}'s Simulation Soundness game to generate proofs for false statements (see Definition 22). Since $\mathbf{A}$ sees at most $q_E \leq q_{P\,\mathrm{NIZK}}$ simulated proofs (namely the ones in the challenge ciphertext) and makes at most $q_D \leq q_{V\,\mathrm{NIZK}}$ decryption queries, it follows from Eq. (4.12) implying

$$\left|\Pr[\mathbf{AG}^3 = \mathtt{win}] - \Pr[\mathbf{AG}^4 = \mathtt{win}]\right| \leq \varepsilon_{\text{NIZK-SS}}.$$

$\mathbf{G}^4 \rightsquigarrow \mathbf{G}^5$**:** Game $\mathbf{G}^5$ is just like $\mathbf{G}^4$, except that for every $l \in \{1, \ldots, |\vec{c}|\}$, ciphertext $c_{l,0}$ of vector $\vec{c} \coloneqq \big((c_{1,0}, c_{1,1}), \ldots, (c_{|\vec{c}|,0}, c_{|\vec{c}|,1})\big)$ of the challenge ciphertext $c^* \coloneqq (p, \mathtt{comm}, \vec{c})$ is now an encryption of $(\tilde{\rho}', \vec{v}_0^*, m^*)$—where $\tilde{\rho}'$ is some sequence of random coins *independent of the one* used by $\Pi_{\mathrm{CS}}.\mathit{Commit}$—instead of $(\rho, \vec{v}_0^*, m^*)$—where ρ is *the sequence of random coins used by* $\Pi_{\mathrm{CS}}.\mathit{Commit}$—similarly to $\mathbf{G}^2$.

By following an argument similar to the one used for $\mathbf{G}^1 \rightsquigarrow \mathbf{G}^2$—where decryption queries are now handled using the alternative decryption key—one can bound the difference in $\mathbf{A}$'s winning advantage for $\mathbf{G}^4$ and for $\mathbf{G}^5$. Since $d_E \leq q_{E\,\mathrm{PKE}}$ and $n \leq n_{\mathrm{PKE}}$, it follows from Eq. (4.11)

$$\left|\Pr[\mathbf{AG}^4 = \mathtt{win}] - \Pr[\mathbf{AG}^5 = \mathtt{win}]\right| \leq \varepsilon_{\text{PKE-IND-CPA}}.$$

$\mathbf{G}^5 \rightsquigarrow \mathbf{G}^6$**:** The only difference between $\mathbf{G}^6$ and $\mathbf{G}^5$ is that in $\mathbf{G}^6$, for each challenge ciphertext $c^* \coloneqq (p, \mathtt{comm}, \vec{c})$, $\mathtt{comm}$ is now a commitment to $(\vec{v}_1^*, m)$ instead of being a commitment to $(\vec{v}_0^*, m)$.

Once again we can reduce distinguishing the two games to breaking the hiding property of the underlying Π_{CS}: the main thing to note is that the sequence of random coins encrypted in each ciphertext in $\vec{c}$ is independent from the sequence used by $\Pi_{\mathrm{CS}}.\mathit{Commit}$. With this in mind, it is easy to see that a reduction analogous to the one used for hop $\mathbf{G}^1 \rightsquigarrow \mathbf{G}^2$ can be used. Since $q_E \leq q_{\mathrm{CS}}$, it

[16] To verify the validity of a ciphertexts's NIZK proof the reduction relies on the $\mathcal{O}_V$ oracle provided by the Simulation Soundness game of the underlying Π_{NIZK}.

follows from Eq. (4.13)

$$\left|\Pr[\mathbf{AG}^4 = \mathtt{win}] - \Pr[\mathbf{AG}^5 = \mathtt{win}]\right| \leq \varepsilon_{\text{CS-Hiding}}.$$

$\mathbf{G}^6 \rightsquigarrow \mathbf{G}^7$**:** The difference between $\mathbf{G}^6$ and $\mathbf{G}^7$ is that in $\mathbf{G}^7$, for every $l \in \{1, \ldots, |\vec{c}|\}$, ciphertext $c_{l,0}$ (of vector $\vec{c} \coloneqq \big((c_{1,0}, c_{1,1}), \ldots, (c_{|\vec{c}|,0}, c_{|\vec{c}|,1})\big)$ of the challenge ciphertext $c^* \coloneqq (p, \mathtt{comm}, \vec{c})$) is an encryption of $(\rho, \vec{v}_1^*, m^*)$—where ρ is *the sequence of random coins used by* $\Pi_{\text{CS}}.\mathit{Commit}$—instead of $(\tilde{\rho}', \vec{v}_0^*, m^*)$—where $\tilde{\rho}'$ is some sequence of random coins, *independent of the one* used by $\Pi_{\text{CS}}.\mathit{Commit}$.

By an argument analogous to the one for $\mathbf{G}^1 \rightsquigarrow \mathbf{G}^2$, we have

$$\left|\Pr[\mathbf{AG}^6 = \mathtt{win}] - \Pr[\mathbf{AG}^7 = \mathtt{win}]\right| \leq \varepsilon_{\text{PKE-IND-CPA}}.$$

$\mathbf{G}^7 \rightsquigarrow \mathbf{G}^8$**:** This game hop is analogous to $\mathbf{G}^2 \rightsquigarrow \mathbf{G}^3$: in $\mathbf{G}^8$, for every $l \in \{1, \ldots, |\vec{c}|\}$, ciphertext $c_{l,0}$ (of vector $\vec{c} \coloneqq \big((c_{1,0}, c_{1,1}), \ldots, (c_{|\vec{c}|,0}, c_{|\vec{c}|,1})\big)$ of the challenge ciphertext $c^* \coloneqq (p, \mathtt{comm}, \vec{c})$) is now encrypted under public key $(v_1^*)_{l,1}$, instead of being encrypted under public key $(v_0^*)_{l,1}$.

Again, by an argument analogous to the one for $\mathbf{G}^1 \rightsquigarrow \mathbf{G}^2$, we have

$$\left|\Pr[\mathbf{AG}^7 = \mathtt{win}] - \Pr[\mathbf{AG}^8 = \mathtt{win}]\right| \leq \varepsilon_{\text{PKE-IK-CPA}}.$$

$\mathbf{G}^8 \rightsquigarrow \mathbf{G}^9$**:** This game hop is analogous to $\mathbf{G}^3 \rightsquigarrow \mathbf{G}^4$: a decryption query for a party B_j—with secret key $((\mathtt{pk}_{j,0}, \mathtt{sk}_{j,0}), (\mathtt{pk}_{j,1}, \mathtt{sk}_{j,1}))$—on a ciphertext $c \coloneqq (p, \mathtt{comm}, \vec{c})$—with $\vec{c} \coloneqq \big((c_{1,0}, c_{1,1}), \ldots, (c_{|\vec{c}|,0}, c_{|\vec{c}|,1})\big)$—behaves slightly differently: rather than decrypting, for $l \in \{1, \ldots, |\vec{c}|\}$, ciphertext $c_{l,1}$ using $\mathtt{sk}_{j,1}$, it returns to decrypting $c_{l,0}$ using $\mathtt{sk}_{j,0}$.

By following an argument analogous to the one for $\mathbf{G}^3 \rightsquigarrow \mathbf{G}^4$, it follows

$$\left|\Pr[\mathbf{AG}^8 = \mathtt{win}] - \Pr[\mathbf{AG}^9 = \mathtt{win}]\right| \leq \varepsilon_{\text{NIZK-SS}}.$$

$\mathbf{G}^9 \rightsquigarrow \mathbf{G}^{10}$**:** This game hop is analogous to $\mathbf{G}^1 \rightsquigarrow \mathbf{G}^2$: for every $l \in \{1, \ldots, |\vec{c}|\}$, ciphertext $c_{l,1}$ (of vector $\vec{c} \coloneqq \big((c_{1,0}, c_{1,1}), \ldots, (c_{|\vec{c}|,0}, c_{|\vec{c}|,1})\big)$ of the challenge ciphertext $c^* \coloneqq (p, \mathtt{comm}, \vec{c})$) is now an encryption of $(\rho, \vec{v}_1^*, m^*)$—where ρ is *the sequence of random coins used by* $\Pi_{\text{CS}}.\mathit{Commit}$—instead of $(\tilde{\rho}, \vec{v}_0^*, m^*)$—where $\tilde{\rho}$ is some sequence of random coins, *independent of the one* used by $\Pi_{\text{CS}}.\mathit{Commit}$.

By following an argument similar to the one for $\mathbf{G}^1 \rightsquigarrow \mathbf{G}^2$, it follows

$$\left|\Pr[\mathbf{AG}^9 = \mathtt{win}] - \Pr[\mathbf{AG}^{10} = \mathtt{win}]\right| \leq \varepsilon_{\text{PKE-IND-CPA}}.$$

$\mathbf{G}^{10} \rightsquigarrow \mathbf{G}_{\mathbf{1}}^{\mathsf{CCA}}$**:** The only difference between the game systems is that the $\mathtt{crs}_{\text{NIZK}}$ output by oracle $\mathcal{O}_S$ returns to the one generated by $\Pi_{\text{NIZK}}.G_{CRS}$, and the NIZK proof p in each challenge ciphertext returns to a real one generated by $\Pi_{\text{NIZK}}.P$.

Taking into account that $\mathbf{A}$ wins $\mathbf{G}_{\mathbf{1}}^{\mathsf{CCA}}$ if it outputs 1 and wins $\mathbf{G}^{10}$ if it outputs 0, by following an argument similar to the one for $\mathbf{G}_{\mathbf{0}}^{\mathsf{CCA}} \rightsquigarrow \mathbf{G}^{1}$, it follows

$$\left|\Pr[\mathbf{AG}^{10} = \mathtt{win}] - \Pr[\mathbf{AG}_{\mathbf{1}}^{\mathsf{CCA}} \neq \mathtt{win}]\right| \leq \varepsilon_{\text{NIZK-ZK}}.$$

□

H.6 Proof of Theorem 5

Follows from a straightforward adaptation of Theorem 4's proof. □

I MDRS-PKE Construction Security Proofs

In this section we give the full security proofs for the MDRS-PKE construction given in Sect. 6.

I.1 Proof of Theorem 6

We prove this result via game hopping.

$\mathbf{G}^{\mathsf{Corr}} \rightsquigarrow \mathbf{G}^{1}$**:** The only difference between games $\mathbf{G}^{1}$ and $\mathbf{G}^{\mathsf{Corr}}$ is that in $\mathbf{G}^{1}$ some decryption queries are handled differently. More concretely, when $\mathcal{O}_D$ is queried on an input $(B_j, c := (\mathtt{vk}, \sigma', c'))$ where c was output by a query $\mathcal{O}_E(A_i, \vec{V}, m)$, $\mathcal{O}_D$ no longer verifies if σ' is a valid signature on c' with respect to $\mathtt{vk}$, and instead simply assumes that it is.

One can reduce distinguishing the two games to breaking the correctness of Π_{DSS}: since the reduction holds all MDVS and PKEBC secret keys and can sign ciphertexts using the $\mathcal{O}_S$ oracle from Π_{DSS}'s $\mathbf{G}^{\mathsf{Corr}}$ game, it can handle any oracle queries. Since $\mathbf{A}$ only makes at most $q_E \leq \min(n_{\text{DSS}}, q_{S\,\text{DSS}})$ and $q_D \leq q_{V\,\text{DSS}}$ queries to $\mathcal{O}_E$ and $\mathcal{O}_D$, respectively, it follows from Equation 6.3 that no adversary $(t_{\text{DSS}}, \varepsilon_{\text{DSS-Corr}})$-breaks the

$$(n_{\text{DSS}}, q_{S\,\text{DSS}}, q_{V\,\text{DSS}})\text{-Correctness}$$

of Π_{DSS}, implying

$$\left|\Pr[\mathbf{AG}^{1} = \mathtt{win}] - \Pr[\mathbf{AG}^{\mathsf{Corr}} = \mathtt{win}]\right| \leq \varepsilon_{\text{DSS-Corr}}.$$

$\mathbf{G}^1 \rightsquigarrow \mathbf{G}^2$**:** The only difference between games $\mathbf{G}^2$ and $\mathbf{G}^1$ is that in $\mathbf{G}^2$ once again some decryption queries are handled differently. More concretely, when $\mathcal{O}_D$ is queried on an input $(B_j, c \coloneqq (\mathtt{vk}, \sigma', c'))$ where c was output by a query $\mathcal{O}_E(A_i, \vec{V}, m)$ such that $B_j \in \vec{V}$, $\mathcal{O}_D$ works as follows: let $(\mathtt{spk}_i, \vec{v}_{\text{MDVS}}, m, \sigma)$ be the plaintext that was encrypted by $\mathit{\Pi}_{\text{PKEBC}}.E$ under $\vec{v}_{\text{PKEBC}}$ (which resulted in ciphertext c'), where $\mathtt{spk}_i$ is A_i's public key,

$$\vec{v}_{\text{MDVS}} \coloneqq (\mathtt{vpk}_{\text{MDVS}1}, \ldots, \mathtt{vpk}_{\text{MDVS}|\vec{v}|}), \text{ and}$$
$$\vec{v}_{\text{PKEBC}} \coloneqq (\mathtt{pk}_{\text{PKEBC}1}, \ldots, \mathtt{pk}_{\text{PKEBC}|\vec{v}|})$$

are, respectively, the vectors of public MDVS verifier keys and public PKEBC receiver keys corresponding to $\vec{V}$, and where

$$\sigma \leftarrow \mathit{\Pi}_{\text{MDVS}}.\mathit{Sig}_{\mathtt{pp}_{\text{MDVS}}}(\mathtt{ssk}_{\text{MDVS}i}, \vec{v}_{\text{MDVS}}, (\vec{v}_{\text{PKEBC}}, m, \mathtt{vk})),$$

is an MDVS signature on $(\vec{v}_{\text{PKEBC}}, m, \mathtt{vk})$, with $\mathtt{ssk}_{\text{MDVS}i}$ being A_i's secret MDVS signing key and $\mathtt{vk}$ being the DSS verification key in c; oracle $\mathcal{O}_D$ no longer decrypts c' using $\mathit{\Pi}_{\text{PKEBC}}.D$ with B_j's PKEBC secret key, and instead simply assumes decryption outputs $(\vec{v}_{\text{PKEBC}}, (\mathtt{spk}_i, \vec{v}_{\text{MDVS}}, m, \sigma))$.

One can reduce distinguishing the two games to breaking the correctness of $\mathit{\Pi}_{\text{PKEBC}}$: since the reduction holds all MDVS and DSS secret keys, has access to the decryption oracle from $\mathit{\Pi}_{\text{PKEBC}}$'s Correctness game, and can access all PKEBC secret keys the adversary may query through the $\mathcal{O}_{SK}$ oracle of $\mathit{\Pi}_{\text{PKEBC}}$'s Correctness game, it can handle any oracle queries. If $\mathbf{A}$ only queries for at most $n_R \leq n_{\text{PKEBC}}$ different receivers, the sum of lengths of the vectors input to $\mathcal{O}_E$ is at most $d_E \leq d_{E\text{PKEBC}}$, and $\mathbf{A}$ makes at most $q_E \leq q_{E\text{PKEBC}}$ and $q_D \leq q_{D\text{PKEBC}}$ queries to oracles $\mathcal{O}_E$ and $\mathcal{O}_D$, respectively, since from Equation 6.1 no adversary $(t_{\text{PKEBC}}, \varepsilon_{\text{PKEBC-Corr}})$-breaks the $(n_{\text{PKEBC}}, d_{E\text{PKEBC}}, q_{E\text{PKEBC}}, q_{D\text{PKEBC}})$-Correctness of $\mathit{\Pi}_{\text{PKEBC}}$, we have

$$\left|\Pr[\mathbf{AG}^2 = \mathtt{win}] - \Pr[\mathbf{AG}^1 = \mathtt{win}]\right| \leq \varepsilon_{\text{PKEBC-Corr}}.$$

$\mathbf{G}^2 \rightsquigarrow \mathbf{G}^3$**:** Game $\mathbf{G}^3$ is just like $\mathbf{G}^2$, except that once again some decryption queries work differently. Essentially, in the same way that $\mathbf{G}^2$ differed from $\mathbf{G}^1$—assuming that the ciphertext c' in each ciphertext $c \coloneqq (\mathtt{vk}, \sigma', c')$ output by a query to $\mathcal{O}_E$ decrypts correctly when $\mathcal{O}_D$ is queried on (B_j, c), B_j being one of the parties in the vector input to $\mathcal{O}_E$—$\mathbf{G}^3$ differs from $\mathbf{G}^2$ in that it assumes that also each MDVS signature σ generated by $\mathcal{O}_E$ using $\mathit{\Pi}_{\text{MDVS}}.\mathit{Sig}$ also verifies as being valid when $\mathcal{O}_D$ is queried on a matching input. To be more precise, for a query $\mathcal{O}_E(A_i, \vec{V}, m)$, let $(\vec{v}_{\text{PKEBC}}, m, \mathtt{vk})$ be the plaintext that was signed by $\mathit{\Pi}_{\text{MDVS}}.\mathit{Sig}$ using $\mathtt{ssk}_i$ and $\vec{v}_{\text{MDVS}}$, where $\mathtt{ssk}_i$ is A_i's secret key,

$$\vec{v}_{\text{MDVS}} \coloneqq (\mathtt{vpk}_{\text{MDVS}1}, \ldots, \mathtt{vpk}_{\text{MDVS}|\vec{v}|}), \text{ and}$$
$$\vec{v}_{\text{PKEBC}} \coloneqq (\mathtt{pk}_{\text{PKEBC}1}, \ldots, \mathtt{pk}_{\text{PKEBC}|\vec{v}|})$$

are, respectively, the vectors of public MDVS verifier keys and public PKEBC receiver keys corresponding to $\vec{V}$, let σ be the resulting signature

$$\sigma \leftarrow \Pi_{\text{MDVS}}.Sig_{\mathtt{pp}_{\text{MDVS}}}(\mathtt{ssk}_{\text{MDVS}i}, \vec{v}_{\text{MDVS}}, (\vec{v}_{\text{PKEBC}}, m, \mathtt{vk})),$$

and let c be the ciphertext output by the $\mathcal{O}_E$ query. Then, when queried on input (B_j, c) such that $B_j \in \vec{V}$, $\mathcal{O}_D$ no longer verifies if σ is valid by running

$$\Pi_{\text{MDVS}}.Vfy(\mathtt{pp}, \mathtt{spk}_i, \mathtt{vsk}_j, \vec{v}_{\text{MDVS}}, (\vec{v}_{\text{PKEBC}}, m, \mathtt{vk}))$$

and instead simply assumes the MDVS signature verification outputs 1.

One can reduce distinguishing the two games to breaking the correctness of the underlying Π_{MDVS} because the MDVS correctness game gives access to a signing oracle, a signature verification oracle and to the secret keys of every signer and verifier (and so the reduction can handle any oracle queries). So, if $\mathbf{A}$ only queries for at most $n_S \leq n_{S\,\text{MDVS}}$ (resp. $n_R \leq n_{V\,\text{MDVS}}$) different sender keys (resp. different receiver keys), makes up to $q_E \leq q_{S\,\text{MDVS}}$ queries to $\mathcal{O}_E$ and up to $q_D \leq q_{V\,\text{MDVS}}$ queries to $\mathcal{O}_D$, and the sum of lengths of the party vectors input to $\mathcal{O}_E$ is at most $d_E \leq d_{S\,\text{MDVS}}$, since from Equation 6.2 no adversary $(\varepsilon_{\text{MDVS-}\mathsf{Corr}}, t_{\text{MDVS}})$-breaks the

$$(n_{S\,\text{MDVS}}, n_{V\,\text{MDVS}}, d_{S\,\text{MDVS}}, q_{S\,\text{MDVS}}, q_{V\,\text{MDVS}})\text{-Correctness}$$

of Π_{MDVS}, it follows

$$\left|\Pr[\mathbf{AG}^3 = \mathtt{win}] - \Pr[\mathbf{AG}^2 = \mathtt{win}]\right| \leq \varepsilon_{\text{MDVS-}\mathsf{Corr}}.$$

To conclude the proof, note that no adversary can win $\mathbf{G}^3$.

I.2 Proof of Theorem 7

We proceed via game hopping.

Algorithm 4 Description of $\mathcal{O}_D$ for game system $\mathbf{G}^1$ in the proof of Theorem 7.

INITIALIZATION
  $\mathtt{vfy}_{\mathrm{DSS}} \leftarrow \emptyset$

$\mathcal{O}_D(B_j, c \coloneqq (\mathtt{vk}, \sigma', c'))$
  if $(\mathtt{vk}, c', \sigma') \notin \mathtt{vfy}_{\mathrm{DSS}}$ **then**
    CONSISTENTDSSVERIFICATION$(\mathtt{vk}, c', \sigma')$
  if $\mathtt{vfy}_{\mathrm{DSS}}[\mathtt{vk}, c', \sigma'] = 0$ **then**
    return $\perp$
  $\big(\vec{v}_{\mathrm{PKEBC}}, (\mathtt{spk} \coloneqq \mathtt{spk}_{\mathrm{MDVS}}, \vec{v}_{\mathrm{MDVS}}, m, \sigma)\big) \leftarrow \Pi_{\mathrm{PKEBC}}.D_{\mathtt{pp}_{\mathrm{PKEBC}}}(\mathtt{rsk}_j.\mathtt{sk}_{\mathrm{PKEBC}}, c')$
  if $\big(\vec{v}_{\mathrm{PKEBC}}, (\mathtt{spk}, \vec{v}_{\mathrm{MDVS}}, m, \sigma)\big) = \perp \ \vee \ |\vec{v}_{\mathrm{PKEBC}}| \neq |\vec{v}_{\mathrm{MDVS}}|$ **then**
    return $\perp$
  $\vec{v} \coloneqq \big((v_{\mathrm{MDVS}1}, v_{\mathrm{PKEBC}1}), \ldots, (v_{\mathrm{MDVS}|\vec{v}_{\mathrm{PKEBC}}|}, v_{\mathrm{PKEBC}|\vec{v}_{\mathrm{PKEBC}}|})\big)$
  if $\nexists\ A \in \mathcal{S}$ with $\mathcal{O}_{SPK}(A) = \mathtt{spk}$ **then**
    return $\perp$
  if $\exists l \in \{1, \ldots, |\vec{v}|\} : (\nexists\ B \in \mathcal{R}$ with $\mathcal{O}_{RPK}(B) = v_l)$ **then**
    return $\perp$
  if $\mathtt{rsk}.\mathtt{rpk} \notin \vec{v}$ **then**
    return $\perp$
  if $\Pi_{\mathrm{MDVS}}.Vfy_{\mathtt{pp}_{\mathrm{MDVS}}}(\mathtt{spk}, \mathtt{rsk}_j.\mathtt{vsk}_{\mathrm{MDVS}}, \vec{v}_{\mathrm{MDVS}}, (\vec{v}_{\mathrm{PKEBC}}, m, \mathtt{vk}), \sigma) \neq \mathtt{valid}$ **then**
    return $\perp$
  return $(\mathtt{spk}, \vec{v}, m)$

CONSISTENTDSSVERIFICATION$(\mathtt{vk}, c', \sigma')$
  $\mathtt{vfy}_{\mathrm{DSS}}[\mathtt{vk}, c', \sigma'] \leftarrow \Pi_{\mathrm{DSS}}.Vfy_{\mathtt{vk}}(c', \sigma')$

$\mathbf{G}^{\mathsf{Cons}} \rightsquigarrow \mathbf{G}^1$**:** The only difference between games $\mathbf{G}^1$ and $\mathbf{G}^{\mathsf{Cons}}$ is that in $\mathbf{G}^1$ decryption queries are handled according to Algorithm 4. Since $\Pi_{\mathrm{DSS}}.Vfy$ is a deterministic algorithm, it follows from

$$\Big|\Pr[\mathbf{AG}^1 = \mathtt{win}] - \Pr[\mathbf{AG}^{\mathsf{Cons}} = \mathtt{win}]\Big| = 0.$$

Algorithm 5 Description of $\mathcal{O}_D$ for game system $\mathbf{G}^2$ in the proof of Theorem 7.

INITIALIZATION
  $\mathtt{dec} \leftarrow \emptyset$ ▷ Write once map: for each $\mathtt{x}$, if $\mathtt{x} \in \mathtt{dec}$, then cannot modify value $\mathtt{dec}[x]$.
  $\mathtt{vfy}_{\mathrm{DSS}} \leftarrow \emptyset$

$\mathcal{O}_D(B_j, c \coloneqq (\mathtt{vk}, \sigma', c'))$
  if $(\mathtt{vk}, c', \sigma') \notin \mathtt{vfy}_{\mathrm{DSS}}$ **then**
    CONSISTENTDSSVERIFICATION$(\mathtt{vk}, c', \sigma')$
  if $\mathtt{vfy}_{\mathrm{DSS}}[\mathtt{vk}, c', \sigma'] = 0$ **then**
    return $\perp$
  if $(\mathtt{rpk}_j.\mathtt{pk}_{\mathrm{PKEBC}}, c') \notin \mathtt{dec}$ **then**
    CONSISTENTDECRYPTION$(B_j, \mathtt{rpk}_j, \mathtt{rsk}_j, c')$
  $\big(\vec{v}_{\mathrm{PKEBC}}, (\mathtt{spk} \coloneqq \mathtt{spk}_{\mathrm{MDVS}}, \vec{v}_{\mathrm{MDVS}}, m, \sigma)\big) \leftarrow \mathtt{dec}[\mathtt{rpk}_j.\mathtt{pk}_{\mathrm{PKEBC}}, c']$
  if $\big(\vec{v}_{\mathrm{PKEBC}}, (\mathtt{spk}, \vec{v}_{\mathrm{MDVS}}, m, \sigma)\big) = \perp \ \vee \ |\vec{v}_{\mathrm{PKEBC}}| \neq |\vec{v}_{\mathrm{MDVS}}|$ **then**
    return $\perp$
  $\vec{v} \coloneqq \big((v_{\mathrm{MDVS}1}, v_{\mathrm{PKEBC}1}), \ldots, (v_{\mathrm{MDVS}|\vec{v}_{\mathrm{PKEBC}}|}, v_{\mathrm{PKEBC}|\vec{v}_{\mathrm{PKEBC}}|})\big)$
  if $\nexists\ A \in \mathcal{S}$ with $\mathcal{O}_{SPK}(A) = \mathtt{spk}$ **then**
    return $\perp$
  if $\exists l \in \{1, \ldots, |\vec{v}|\} : (\nexists\ B \in \mathcal{R}$ with $\mathcal{O}_{RPK}(B) = v_l)$ **then**
    return $\perp$
  if $\mathtt{rsk}.\mathtt{rpk} \notin \vec{v}$ **then**
    return $\perp$
  if $\mathit{\Pi}_{\mathrm{MDVS}}.\mathit{Vfy}_{\mathtt{ppMDVS}}(\mathtt{spk}, \mathtt{vsk}_{\mathrm{MDVS}}, \vec{v}_{\mathrm{MDVS}}, (\vec{v}_{\mathrm{PKEBC}}, m, \mathtt{vk}), \sigma) \neq \mathtt{valid}$ **then**
    return $\perp$
  return $(\mathtt{spk}, \vec{v}, m)$

CONSISTENTDECRYPTION$(B_j, \mathtt{rpk}_j, \mathtt{rsk}_j, c')$
  $\big(\vec{v}_{\mathrm{PKEBC}}, (\mathtt{spk}, \vec{v}_{\mathrm{MDVS}}, m, \sigma)\big) \leftarrow \mathit{\Pi}_{\mathrm{PKEBC}}.D_{\mathtt{ppPKEBC}}(\mathtt{rsk}_j.\mathtt{sk}_{\mathrm{PKEBC}}, c')$
  if $\big(\vec{v}_{\mathrm{PKEBC}}, (\mathtt{spk}, \vec{v}_{\mathrm{MDVS}}, m, \sigma)\big) = \perp$ **then** ▷ Decryption failed.
    $\mathtt{dec}[\mathtt{rpk}_j.\mathtt{pk}_{\mathrm{PKEBC}}, c'] \leftarrow \perp$
  else
    For each $\mathtt{pk} \in \vec{v}_{\mathrm{PKEBC}}$: $\mathtt{dec}[\mathtt{pk}, c'] \leftarrow \big(\vec{v}_{\mathrm{PKEBC}}, (\mathtt{spk} \coloneqq \mathtt{spk}_{\mathrm{MDVS}}, \vec{v}_{\mathrm{MDVS}}, m, \sigma)\big)$

CONSISTENTDSSVERIFICATION$(\mathtt{vk}, c', \sigma')$
  $\mathtt{vfy}_{\mathrm{DSS}}[\mathtt{vk}, c', \sigma'] \leftarrow \mathit{\Pi}_{\mathrm{DSS}}.\mathit{Vfy}_{\mathtt{vk}}(c', \sigma')$

$\mathbf{G}^1 \rightsquigarrow \mathbf{G}^2$: The only difference between games $\mathbf{G}^2$ and $\mathbf{G}^1$ is that in $\mathbf{G}^2$ decryption queries are handled according to Algorithm 5. Note that in $\mathbf{G}^2$ it is assumed that for any x, if $x \in \mathtt{dec}$ then $\mathtt{dec}[x]$ is not modified.

One can reduce distinguishing $\mathbf{G}^2$ and $\mathbf{G}^1$ or breaking the aforementioned assumption (that for any x, if $x \in \mathtt{dec}$ then $\mathtt{dec}[x]$ is not modified) to breaking the consistency of the underlying $\mathit{\Pi}_{\mathrm{PKEBC}}$ because the reduction can query for the PKEBC secret key of any receiver that the adversary may query for. If $\mathbf{A}$ only queries $\mathcal{O}_{RPK}$, $\mathcal{O}_{RK}$, $\mathcal{O}_E$ and $\mathcal{O}_D$ on at most $n_R \leq n_{\mathrm{PKEBC}}$ different receivers, makes at most $q_E \leq q_{E\mathrm{PKEBC}}$ and $q_D \leq q_{D\mathrm{PKEBC}}$ queries to $\mathcal{O}_E$ and $\mathcal{O}_D$, respectively, and the sum of lengths of the party vectors $\mathbf{A}$ inputs to $\mathcal{O}_E$ is at most $d_E \leq d_{E\mathrm{PKEBC}}$, then it follows from Equation 6.4 that

$$\Big|\Pr[\mathbf{AG}^2 = \mathtt{win}] - \Pr[\mathbf{AG}^1 = \mathtt{win}]\Big| \leq \varepsilon_{\mathrm{PKEBC\text{-}Cons}}.$$

Algorithm 6 Description of $\mathcal{O}_D$ for game system $\mathbf{G}^3$ in the proof of Theorem 7.

INITIALIZATION
  $\mathtt{vfy}_{\mathrm{MDVS}} \leftarrow \emptyset$
  $\mathtt{dec} \leftarrow \emptyset$ ▷ Write once map: for each $\mathtt{x}$, if $\mathtt{x} \in \mathtt{dec}$, then cannot modify value $\mathtt{dec}[x]$.
  $\mathtt{vfy}_{\mathrm{DSS}} \leftarrow \emptyset$

$\mathcal{O}_D(B_j, c \coloneqq (\mathtt{vk}, \sigma', c'))$
  if $(\mathtt{vk}, c', \sigma') \notin \mathtt{vfy}_{\mathrm{DSS}}$ **then**
    ConsistentDSSVerification$(\mathtt{vk}, c', \sigma')$
  if $\mathtt{vfy}_{\mathrm{DSS}}[\mathtt{vk}, c', \sigma'] = 0$ **then**
    return $\perp$
  if $(\mathtt{rpk}_j.\mathtt{pk}_{\mathrm{PKEBC}}, c') \notin \mathtt{dec}$ **then**
    ConsistentDecryption$(B_j, \mathtt{rpk}_j, \mathtt{rsk}_j, c')$
  $\big(\vec{v}_{\mathrm{PKEBC}}, (\mathtt{spk} \coloneqq \mathtt{spk}_{\mathrm{MDVS}}, \vec{v}_{\mathrm{MDVS}}, m, \sigma)\big) \leftarrow \mathtt{dec}[\mathtt{rpk}_j.\mathtt{pk}_{\mathrm{PKEBC}}, c']$
  if $\big(\vec{v}_{\mathrm{PKEBC}}, (\mathtt{spk}, \vec{v}_{\mathrm{MDVS}}, m, \sigma)\big) = \perp \ \vee \ |\vec{v}_{\mathrm{PKEBC}}| \neq |\vec{v}_{\mathrm{MDVS}}|$ **then**
    return $\perp$
  $\vec{v} \coloneqq \big((v_{\mathrm{MDVS}_1}, v_{\mathrm{PKEBC}_1}), \ldots, (v_{\mathrm{MDVS}_{|\vec{v}_{\mathrm{PKEBC}}|}}, v_{\mathrm{PKEBC}_{|\vec{v}_{\mathrm{PKEBC}}|}})\big)$
  if $\nexists\ A \in \mathcal{S}$ with $\mathcal{O}_{SPK}(A) = \mathtt{spk}$ **then**
    return $\perp$
  if $\exists l \in \{1, \ldots, |\vec{v}|\} : (\nexists\ B \in \mathcal{R}$ with $\mathcal{O}_{RPK}(B) = v_l)$ **then**
    return $\perp$
  if $\mathtt{rsk}.\mathtt{rpk} \notin \vec{v}$ **then**
    return $\perp$
  if $\nexists$ query $\mathcal{O}_{RK}(B_j)$ **then** ▷ B_j's secret key was not queried.
    if $\big(\mathtt{spk}, \vec{v}_{\mathrm{MDVS}}, (\vec{v}_{\mathrm{PKEBC}}, m, \mathtt{vk}), \sigma\big) \notin \mathtt{vfy}_{\mathrm{MDVS}}$ **then**
      ConsistentMDVSVerification$(\mathtt{rsk}_j.\mathtt{vsk}, \mathtt{spk}, \vec{v}_{\mathrm{MDVS}}, (\vec{v}_{\mathrm{PKEBC}}, m, \mathtt{vk}), \sigma)$
    if $\mathtt{vfy}_{\mathrm{MDVS}}[\mathtt{spk}, \vec{v}_{\mathrm{MDVS}}, m, \sigma] = 0$ **then**
      return $\perp$
  else ▷ $\exists$ query $\mathcal{O}_{RK}(B_j)$
    if $\Pi_{\mathrm{MDVS}}.Vfy_{\mathtt{pp}_{\mathrm{MDVS}}}(\mathtt{spk}, \mathtt{rsk}_j.\mathtt{vsk}, \vec{v}_{\mathrm{MDVS}}, m, \sigma) = 0$ **then**
      return $\perp$
  return $(\mathtt{spk}, \vec{v}, m)$

ConsistentMDVSVerification$(\mathtt{vsk}, \mathtt{spk}, \vec{v}_{\mathrm{MDVS}}, m, \sigma)$
  $\mathtt{vfy}_{\mathrm{MDVS}}[\mathtt{spk}, \vec{v}_{\mathrm{MDVS}}, m, \sigma] \leftarrow \Pi_{\mathrm{MDVS}}.Vfy_{\mathtt{pp}_{\mathrm{MDVS}}}(\mathtt{spk}, \mathtt{vsk}, \vec{v}_{\mathrm{MDVS}}, m, \sigma)$

ConsistentDecryption$(B_j, \mathtt{rpk}_j, \mathtt{rsk}_j, c')$
  $\big(\vec{v}_{\mathrm{PKEBC}}, (\mathtt{spk}, \vec{v}_{\mathrm{MDVS}}, m, \sigma)\big) \leftarrow \Pi_{\mathrm{PKEBC}}.D_{\mathtt{pp}_{\mathrm{PKEBC}}}(\mathtt{rsk}_j.\mathtt{sk}_{\mathrm{PKEBC}}, c')$
  if $\big(\vec{v}_{\mathrm{PKEBC}}, (\mathtt{spk}, \vec{v}_{\mathrm{MDVS}}, m, \sigma)\big) = \perp$ **then** ▷ Decryption failed.
    $\mathtt{dec}[\mathtt{rpk}_j.\mathtt{pk}_{\mathrm{PKEBC}}, c'] \leftarrow \perp$
  else
    For each $\mathtt{pk} \in \vec{v}_{\mathrm{PKEBC}}$: $\mathtt{dec}[\mathtt{pk}, c'] \leftarrow \big(\vec{v}_{\mathrm{PKEBC}}, (\mathtt{spk} \coloneqq \mathtt{spk}_{\mathrm{MDVS}}, \vec{v}_{\mathrm{MDVS}}, m, \sigma)\big)$

ConsistentDSSVerification$(\mathtt{vk}, c', \sigma')$
  $\mathtt{vfy}_{\mathrm{DSS}}[\mathtt{vk}, c', \sigma'] \leftarrow \Pi_{\mathrm{DSS}}.Vfy_{\mathtt{vk}}(c', \sigma')$

$\mathbf{G}^2 \rightsquigarrow \mathbf{G}^3$: The only difference between games $\mathbf{G}^3$ and $\mathbf{G}^2$ is that in $\mathbf{G}^3$ decryption queries are handled according to Algorithm 6.

Considering the consistency notions for MDVS schemes (Definition 24) and for MDRS-PKE schemes (Definition 7), it is easy to see that one can reduce distinguishing the two games to breaking the consistency of the underlying MDVS scheme Π_{MDVS}: note that the reduction holds all PKEBC secret keys, all dishonest receiver's MDVS key-pairs and all senders MDVS key-pairs, and that it can rely on the $\mathcal{O}_V$ oracle of provided by Π_{MDVS}'s consistency game $\mathbf{G}^{\mathsf{Cons}}$ to handle decryption queries. It then follows from Equation 6.5 that if $\mathbf{A}$ only

queries for at most $n_S \leq n_{S\text{MDVS}}$ (resp. $n_R \leq n_{V\text{MDVS}}$) different sender keys (resp. different receiver keys), makes up to $q_E \leq q_{S\text{MDVS}}$ queries to $\mathcal{O}_E$ and up to $q_D \leq q_{V\text{MDVS}}$ queries to $\mathcal{O}_D$, and the sum of lengths of the party vectors input to $\mathcal{O}_E$ is at most $d_E \leq d_{S\text{MDVS}}$, then

$$\left|\Pr[\mathbf{AG}^3 = \texttt{win}] - \Pr[\mathbf{AG}^2 = \texttt{win}]\right| \leq \varepsilon_{\text{MDVS-Cons}}.$$

To conclude the proof, note that the probability of an adversary $\mathbf{A}$ to win $\mathbf{G}^3$ is 0, implying:

$$\Pr[\mathbf{AG}^3 = \texttt{win}] = 0.$$

I.3 Proof of Theorem 8

Proof. This proof proceeds via a sequence of games.

$\mathbf{G}^{\mathsf{Unforg}} \rightsquigarrow \mathbf{G}^1$: The difference between $\mathbf{G}^1$ and $\mathbf{G}^{\mathsf{Unforg}}$ is that in $\mathbf{G}^1$, when $\mathcal{O}_D$ is queried on an input $(B_j, c \coloneqq (\texttt{vk}, \sigma', c'))$ such that there is a query $\mathcal{O}_E(A_i, \vec{V}, m)$ that output $c^* \coloneqq (\texttt{vk}^*, \sigma'^*, c'^*)$ with $c \neq c^*$ and $\texttt{vk} = \texttt{vk}^*$, $\mathcal{O}_D$ simply outputs $\perp$.

One can reduce distinguishing the two games to breaking the 1-sEUF-CMA security of Π_{DSS}: the reduction holds all MDVS and PKEBC secret keys and can sign ciphertexts using the $\mathcal{O}_S$ oracle from Π_{DSS}'s $\mathbf{G}^{1\text{-sEUF-CMA}}$ game so it can handle any oracle queries. If $\mathbf{A}$ only makes at most $q_E \leq \min(n_{\text{DSS}}, q_{S\text{DSS}})$ and $q_D \leq q_{V\text{DSS}}$ queries to $\mathcal{O}_E$ and $\mathcal{O}_D$, respectively, since from Equation 6.8 no adversary $(t_{\text{DSS}}, \varepsilon_{\text{DSS-1-sEUF-CMA}})$-breaks the

$$(n_{\text{DSS}}, q_{S\text{DSS}}, q_{V\text{DSS}})\text{-1-sEUF-CMA}$$

of Π_{DSS}, it follows

$$\left|\Pr[\mathbf{AG}^1 = \texttt{win}] - \Pr[\mathbf{AG}^{\mathsf{Unforg}} = \texttt{win}]\right| \leq \varepsilon_{\text{DSS-1-sEUF-CMA}}.$$

We can now directly reduce to the Unforgeability game of Π_{MDVS}. To see why, note that $\mathbf{G}^1$ already outputs $\perp$ for any query $\mathcal{O}_D(B_j, c \coloneqq (\texttt{vk}, \sigma', c'))$ such that there was a query $\mathcal{O}_E(A_i, \vec{V}, m)$ that output $c^* \coloneqq (\texttt{vk}^*, \sigma'^*, c'^*)$ with $c \neq c^*$ and $\texttt{vk} = \texttt{vk}^*$. At this point we then only have to argue that for the remainder of the $\mathcal{O}_D$ queries either the adversary does not win the MDRS-PKE unforgeability game or a reduction to the unforgeability game of the underlying Π_{MDVS} would win. So, for any $\mathcal{O}_D(B_j, c \coloneqq (\texttt{vk}, \sigma', c'))$ query it only remains to consider the two following cases:

- at least one prior $\mathcal{O}_E$ query output $c^* = c$; or
- all prior $\mathcal{O}_E(A_i, \vec{V}, m)$ queries output some $c^* \coloneqq (\texttt{vk}^*, \sigma'^*, c'^*)$ with $\texttt{vk}^* \neq \texttt{vk}$.

We begin with the latter case; in this case if $\mathcal{O}_D$ outputs something other than $\perp$ then the MDVS signature encrypted by c' verified as being valid on a triple $(\vec{v}_{\text{PKEBC}}, m, \texttt{vk})$ which was never signed (because $\texttt{vk}$ was not output as part

of any $\mathcal{O}_E$ ciphertext). Therefore in this case we can reduce to the underlying Π_{MDVS}'s unforgeability.

Now consider the former case: at least one prior $\mathcal{O}_E$ query output $c^* = c$. Let $(\texttt{spk}_i, \vec{v}, m) \neq \perp$ be the output of the $\mathcal{O}_D(B_j, c)$ query above. (Note that $\mathcal{O}_D$ queries outputting $\perp$ do not allow the adversary to win the MDRS-PKE unforgeability game.) Again, we proceed by case distinction:

Case I There is a sender/receiver-vector pair $(A_i', \vec{V}')$ matching $(\texttt{spk}_i, \vec{v})$ for which there was no prior query $\mathcal{O}_E(A_i', \vec{V}', m)$.
In this case one can use the adversary to win the unforgeability game of the underlying MDVS scheme: a reduction just has to keep track of the inputs and outputs of the $\mathcal{O}_E$ queries and then, upon a query $\mathcal{O}_D(B_j, c)$, use this information to make a query to $\mathcal{O}_V$ to verify a signature on a message m with respect to a sender A_i, a receiver vector $\vec{V}$ (with $B_j \in \vec{V}$) using B_j as the verifier, such that there was no prior query to $\mathcal{O}_E$ on $(A_i, \vec{V}, m)$.

Case II There is no sender/receiver-vector pair $(A_i', \vec{V}')$ matching $(\texttt{spk}_i, \vec{v})$ for which there was no prior query $\mathcal{O}_E(A_i', \vec{V}', m)$.
This means for every sender/receiver-vector pair $(A_i', \vec{V}')$ matching $(\texttt{spk}_i, \vec{v})$ there was a query $\mathcal{O}_E(A_i', \vec{V}', m)$. So, in this case the adversary does not win the unforgeability game of the MDRS-PKE.

Since from Equation 6.7 no adversary $(\varepsilon_{\text{MDVS-Unforg}}, t_{\text{MDVS}})$-breaks the

$$(n_{S\,\text{MDVS}}, n_{V\,\text{MDVS}}, d_{S\,\text{MDVS}}, q_{S\,\text{MDVS}}, q_{V\,\text{MDVS}})\text{-Unforgeability}$$

of Π_{MDVS}, if $\mathbf{A}$ only queries for at most $n_S \leq n_{S\,\text{MDVS}}$ (resp. $n_R \leq n_{V\,\text{MDVS}}$) different sender keys (resp. different receiver keys), makes up to $q_E \leq q_{S\,\text{MDVS}}$ queries to $\mathcal{O}_E$ and up to $q_D \leq q_{V\,\text{MDVS}}$ queries to $\mathcal{O}_D$, and the sum of lengths of the party vectors input to $\mathcal{O}_E$ is at most $d_E \leq d_{S\,\text{MDVS}}$, it follows

$$\Pr[\mathbf{AG}^1 = \texttt{win}] \leq \varepsilon_{\text{MDVS-Unforg}}.$$

□

I.4 Proof of Theorem 9

Theorem 9 follows from a straightforward adaptation Theorem 10's proof. □

I.5 Proof of Theorem 10

Ciphertexts of our MDRS-PKE scheme are triples $c \coloneqq (\texttt{vk}, \sigma', c')$ where $\texttt{vk}$ and σ' are, respectively, a verification key and a signature of the underlying Π_{DSS}, and c' is a Π_{PKEBC} ciphertext. At a high level, our goal is to reduce an adversary from distinguishing MDRS-PKE's $\mathbf{G}_\mathbf{0}^{\text{IK-CCA-2}}$ and $\mathbf{G}_\mathbf{1}^{\text{IK-CCA-2}}$ games to one distinguishing the analogous games for (the underlying) Π_{PKEBC}. If the adversary makes a decryption query

$$\mathcal{O}_D(B_j, c \coloneqq (\texttt{vk}, \sigma', c'))$$

where c was output by a challenge query to $\mathcal{O}_E$ the reduction can simply output `test` as this is a disallowed query; if c was not output by a query to $\mathcal{O}_E$ and the PKEBC ciphertext c' of c was also not output (as part of any ciphertext output by $\mathcal{O}_E$) then we can use the decryption oracle $\mathcal{O}_D$ of the IK-CCA-2 games of Π_{PKEBC}. However there is a problem when c was not output by any challenge query but its c' component was because this disallows us from using the decryption oracle $\mathcal{O}_D$ of Π_{PKEBC}'s IK-CCA-2 security games. To get around this we will show—via a sequence of hybrids starting from $\mathbf{G}_\beta^{\text{IK-CCA-2}}$ and ending in $\mathbf{G}_\beta^5$, for $\beta \in \{0,1\}$—that such queries can be handled by simply outputting $\perp$, thus enabling a reduction to the IK-CCA-2 security of the underlying Π_{PKEBC} scheme. Consider any query $\mathcal{O}_D(B_j, c \coloneqq (\mathtt{vk}, \sigma', c'))$ such that c was not output by a challenge query to $\mathcal{O}_E$ but c' was (in the sense above); the following hybrids highlight the main steps of the proof:

$\mathbf{G}_\beta^1$: if any $\mathcal{O}_E$ query output $c^* \coloneqq (\mathtt{vk}^*, \sigma'^*, c'^*)$ with $\mathtt{vk} = \mathtt{vk}^*$, $\mathcal{O}_D$ outputs $\perp$;

$\mathbf{G}_\beta^2$: if any query $\mathcal{O}_E\big((A_{i,0}, \vec{V_0}), (A_{i,1}, \vec{V_1}), m\big)$ output $c^* \coloneqq (\mathtt{vk}^*, \sigma'^*, c'^*)$ with $B_j \notin \vec{V_\beta}$ (and $\mathtt{vk} \neq \mathtt{vk}^*$), $\mathcal{O}_D$ outputs $\perp$;

$\mathbf{G}_\beta^4$: if any query $\mathcal{O}_E\big((A_{i,0}, \vec{V_0}), (A_{i,1}, \vec{V_1}), m\big)$ output $c^* \coloneqq (\mathtt{vk}^*, \sigma'^*, c'^*)$ with $B_j \in \vec{V_\beta}$ and $\mathtt{vk} \neq \mathtt{vk}^*$, $\mathcal{O}_D$ outputs $\perp$.

In $\mathbf{G}_\beta^4$ we know how to handle any decryption queries for ciphertexts that were not output by $\mathcal{O}_E$ but whose PKEBC component was, and hence we are (essentially) set to make the final reduction to the IK-CCA-2 security of Π_{PKEBC}. In the following, let $\beta \in \{0,1\}$.

$\mathbf{G}_\beta^{\text{IK-CCA-2}} \rightsquigarrow \mathbf{G}_\beta^1$: The difference between $\mathbf{G}_\beta^1$ and $\mathbf{G}_\beta^{\text{IK-CCA-2}}$ is that in $\mathbf{G}_\beta^1$ some decryption queries are handled differently: when $\mathcal{O}_D$ is queried on an input $(B_j, c \coloneqq (\mathtt{vk}, \sigma', c'))$ such that there is a query $\mathcal{O}_E\big((A_{i,0}, \vec{V_0}), (A_{i,1}, \vec{V_1}), m\big)$ that output $c^* \coloneqq (\mathtt{vk}^*, \sigma'^*, c'^*)$ with $c \neq c^*$, $c' = c'^*$ and $\mathtt{vk} = \mathtt{vk}^*$, $\mathcal{O}_D$ simply outputs $\perp$.

One can reduce distinguishing the two games to breaking the 1-sEUF-CMA security of Π_{DSS}: since the reduction holds all MDVS and PKEBC secret keys and can sign ciphertexts using oracle $\mathcal{O}_S$ from Π_{DSS}'s $\mathbf{G}^{\text{1-sEUF-CMA}}$ game it can handle any oracle queries. If $\mathbf{A}$ only makes at most $q_E \leq \min(n_{\text{DSS}}, q_{S\,\text{DSS}})$ and $q_D \leq q_{V\,\text{DSS}}$ queries to $\mathcal{O}_E$ and $\mathcal{O}_D$, respectively, since from Equation 6.11 no adversary $(t_{\text{DSS}}, \varepsilon_{\text{DSS-1-sEUF-CMA}})$-breaks the

$$(n_{\text{DSS}}, q_{S\,\text{DSS}}, q_{V\,\text{DSS}})\text{-1-sEUF-CMA}$$

of Π_{DSS}, it follows

$$\Big|\Pr[\mathbf{AG}_\beta^1 = \mathtt{win}] - \Pr[\mathbf{AG}_\beta^{\text{IK-CCA-2}} = \mathtt{win}]\Big| \leq \varepsilon_{\text{DSS-1-sEUF-CMA}}.$$

$\mathbf{G}^1_\beta \rightsquigarrow \mathbf{G}^2_\beta$**:** In $\mathbf{G}^2_\beta$ some decryption queries are once again handled differently; when $\mathcal{O}_D$ is queried on an input $(B_j, c \coloneqq (\mathtt{vk}, \sigma', c'))$ and there is a query $\mathcal{O}_E\big((A_{i,0}, \vec{V_0}), (A_{i,1}, \vec{V_1}), m\big)$ that output $c^* \coloneqq (\mathtt{vk}^*, \sigma'^*, c'^*)$ with $c \neq c^*$, $c' = c'^*$, $\mathtt{vk} \neq \mathtt{vk}^*$, and $B_j \notin \vec{V_\beta}$, $\mathcal{O}_D$ outputs $\perp$.

One can reduce distinguishing $\mathbf{G}^2_\beta$ and $\mathbf{G}^1_\beta$ to breaking Π_{PKEBC}'s robustness (as defined in Definition 2). The main things to note are:

1. a reduction to Π_{PKEBC}'s robustness can access the secret keys of any party whose keys the adversary may query for, and can rely on oracle $\mathcal{O}_D$ provided by Π_{PKEBC}'s robustness game to handle decryption queries;
2. for a query $\mathcal{O}_E\big((A_{i,0}, \vec{V_0}), (A_{i,1}, \vec{V_1}), m\big)$, the reduction can make a query $\mathcal{O}_E(A_{i,\beta}, \vec{V_\beta}, m)$ to the robustness game of Π_{PKEBC}; and
3. if a query $\mathcal{O}_D(B_j, c \coloneqq (\mathtt{vk}, \sigma', c'))$ does not output $\perp$, then the decryption by B_j of the PKEBC ciphertext c' (that is part of c) did not result in $\perp$ either.

If $\mathbf{A}$ only queries for at most $n_R \leq n_{\text{PKEBC}}$ different receivers, the sum of lengths of the vectors input to $\mathcal{O}_E$ is at most $d_E \leq d_{E\text{PKEBC}}$, and $\mathbf{A}$ makes at most $q_E \leq q_{E\text{PKEBC}}$ and $q_D \leq q_{D\text{PKEBC}}$ queries to oracles $\mathcal{O}_E$ and $\mathcal{O}_D$, from Equation 6.9 it then follows

$$\Big|\Pr[\mathbf{AG}^2_\beta = \mathtt{win}] - \Pr[\mathbf{AG}^1_\beta = \mathtt{win}]\Big| \leq \varepsilon_{\text{PKEBC-Rob}}.$$

$\mathbf{G}^2_\beta \rightsquigarrow \mathbf{G}^3_\beta$**:** In $\mathbf{G}^3_\beta$ some decryption queries are handled differently: when $\mathcal{O}_D$ is queried on an input $(B_j, c \coloneqq (\mathtt{vk}, \sigma', c'))$ such that there is a query $\mathcal{O}_E\big((A_{i,0}, \vec{V_0}), (A_{i,1}, \vec{V_1}), m\big)$ that output $c^* \coloneqq (\mathtt{vk}^*, \sigma'^*, c'^*)$ with $c \neq c^*$, $\mathtt{vk} \neq \mathtt{vk}^*$, $c' = c'^*$, and $B_j \in \vec{V_\beta}$, $\mathcal{O}_D$ works as follows: let

$$(\mathtt{spk}_{i,\beta}, \vec{v_\beta}_{\text{MDVS}}, m, \sigma)$$

be the plaintext that was encrypted by $\Pi_{\text{PKEBC}}.E$ under $\vec{v_\beta}_{\text{PKEBC}}$ (which resulted in ciphertext c'), where $\mathtt{spk}_{i,\beta}$ is $A_{i,\beta}$'s public key,

$$\vec{v_\beta}_{\text{MDVS}} \coloneqq (\mathtt{vpk}_{\text{MDVS}1,\beta}, \ldots, \mathtt{vpk}_{\text{MDVS}|\vec{v}|,\beta}), \text{ and}$$
$$\vec{v_\beta}_{\text{PKEBC}} \coloneqq (\mathtt{pk}_{\text{PKEBC}1,\beta}, \ldots, \mathtt{pk}_{\text{PKEBC}|\vec{v}|,\beta})$$

are, respectively, the vectors of public MDVS verifier keys and public PKEBC receiver keys corresponding to $\vec{V_\beta}$, and where

$$\sigma \leftarrow \Pi_{\text{MDVS}}.Sig_{\mathtt{pp}_{\text{MDVS}}}(\mathtt{ssk}_{\text{MDVS}i,\beta}, \vec{v_\beta}_{\text{MDVS}}, (\vec{v_\beta}_{\text{PKEBC}}, m, \mathtt{vk})),$$

is an MDVS signature on $(\vec{v_\beta}_{\text{PKEBC}}, m, \mathtt{vk})$, with $\mathtt{ssk}_{\text{MDVS}i,\beta}$ being $A_{i,\beta}$'s secret MDVS signing key and $\mathtt{vk}$ being the DSS verification key in c; if the signature σ' verifies as valid for c' under verification key $\mathtt{vk}$, oracle $\mathcal{O}_D$ no longer decrypts c' using $\Pi_{\text{PKEBC}}.D$ with B_j's PKEBC secret key, and instead simply assumes decryption outputs

$$(\vec{v_\beta}_{\text{PKEBC}}, (\mathtt{spk}_{i,\beta}, \vec{v_\beta}_{\text{MDVS}}, m, \sigma)).$$

By considering a reduction and an argument similar to the one in the proof of Theorem 6 for the step $\mathbf{G}^1 \rightsquigarrow \mathbf{G}^2$,[17] we have

$$\left|\Pr[\mathbf{AG}^3_\beta = \mathtt{win}] - \Pr[\mathbf{AG}^2_\beta = \mathtt{win}]\right| \leq \varepsilon_{\text{PKEBC-Corr}}.$$

$\mathbf{G}^3_\beta \rightsquigarrow \mathbf{G}^4_\beta$: In $\mathbf{G}^4_\beta$ some decryption queries are handled differently: when $\mathcal{O}_D$ is queried on an input $(B_j, c \coloneqq (\mathtt{vk}, \sigma', c'))$ such that there is a query $\mathcal{O}_E\big((A_{i,0}, \vec{V_0}), (A_{i,1}, \vec{V_1}), m\big)$ that output $c^* \coloneqq (\mathtt{vk}^*, \sigma'^*, c'^*)$ with $c \neq c^*$, $c' = c'^*$, $\mathtt{vk} \neq \mathtt{vk}^*$, and $B_j \in \vec{V_\beta}$, $\mathcal{O}_D$ simply outputs $\perp$. Before moving to showing that

$$\left|\Pr[\mathbf{AG}^4_\beta = \mathtt{win}] - \Pr[\mathbf{AG}^3_\beta = \mathtt{win}]\right| \leq \varepsilon_{\text{MDVS-Unforg}}$$

(by explaining why one can make a reduction to the Unforgeability notion of the underlying Π_{MDVS} scheme), we want to note that, at this point, for any $\mathcal{O}_D(B_j, c \coloneqq (\mathtt{vk}, \sigma', c'))$ query the adversary makes such that there is a query to $\mathcal{O}_E$ that output $c^* \coloneqq (\mathtt{vk}^*, \sigma'^*, c'^*)$ with $c \neq c^*$ and $c' = c'^*$, $\mathcal{O}_D$ simply outputs $\perp$ (i.e. we are essentially set to make the reduction to the IK-CCA-2 security of the underlying Π_{PKEBC} scheme).

$\mathbf{G}^3_\beta$ already outputs $\perp$ for any query $\mathcal{O}_D(B_j, c \coloneqq (\mathtt{vk}, \sigma', c'))$ such that:

- there was a query $\mathcal{O}_E\big((A_{i,0}, \vec{V_0}), (A_{i,1}, \vec{V_1}), m\big)$ that output $c^* \coloneqq (\mathtt{vk}^*, \sigma'^*, c'^*)$ with $c \neq c^*$, $c' = c'^*$ and $\mathtt{vk} = \mathtt{vk}^*$ (see $\mathbf{G}^{\text{IK-CCA-2}}_\beta \rightsquigarrow \mathbf{G}^1_\beta$); or
- there was a query $\mathcal{O}_E\big((A_{i,0}, \vec{V_0}), (A_{i,1}, \vec{V_1}), m\big)$ that output $c^* \coloneqq (\mathtt{vk}^*, \sigma'^*, c'^*)$ with $c \neq c^*$, $c' = c'^*$, $\mathtt{vk} \neq \mathtt{vk}^*$ and $B_j \notin \vec{V_\beta}$ (see $\mathbf{G}^1_\beta \rightsquigarrow \mathbf{G}^2_\beta$).

Thus we only have to make sure that we can use the adversary to break the unforgeability of Π_{MDVS} for the remainder of the queries, i.e. queries to $\mathcal{O}_D(B_j, c \coloneqq (\mathtt{vk}, \sigma', c'))$ such that:

1. there is no query $\mathcal{O}_E\big((A_{i,0}, \vec{V_0}), (A_{i,1}, \vec{V_1}), m\big)$ that output $c^* \coloneqq (\mathtt{vk}^*, \sigma'^*, c'^*)$ with $c \neq c^*$, $c' = c'^*$ and $\mathtt{vk} = \mathtt{vk}^*$; and
2. there is no query $\mathcal{O}_E\big((A_{i,0}, \vec{V_0}), (A_{i,1}, \vec{V_1}), m\big)$ that output $c^* \coloneqq (\mathtt{vk}^*, \sigma'^*, c'^*)$ with $c \neq c^*$, $c' = c'^*$, $\mathtt{vk} \neq \mathtt{vk}^*$ and $B_j \notin \vec{V_\beta}$; and
3. there is a query $\mathcal{O}_E\big((A_{i,0}, \vec{V_0}), (A_{i,1}, \vec{V_1}), m\big)$ that output $c^* \coloneqq (\mathtt{vk}^*, \sigma'^*, c'^*)$ with $c \neq c^*$, $c' = c'^*$, $\mathtt{vk} \neq \mathtt{vk}^*$ and $B_j \in \vec{V_\beta}$.

Note that since $B_j \in \vec{V_\beta}$ and because we are assuming the correctness of Π_{PKEBC}, the reduction does not need to attempt to decrypt c', and instead can simply assume that the decryption is

$$(\vec{v_\beta}_{\text{PKEBC}}, (\mathtt{spk}_{i,\beta}, \vec{v_\beta}_{\text{MDVS}}, m, \sigma)),$$

[17] The reduction has to be slightly modified: for query $\mathcal{O}_E\big((A_{i,0}, \vec{V_0}), (A_{i,1}, \vec{V_1}), m\big)$ the reduction makes a query $\mathcal{O}_E(A_{i,\beta}, \vec{V_\beta}, m)$ to Π_{PKEBC}'s Correctness game.

as explained in step $\mathbf{G}^2_\beta \rightsquigarrow \mathbf{G}^3_\beta$—this is necessary because we need the MDVS keys the reduction obtains from the decryption of c' to match the ones from the underlying $\mathbf{G}^{\mathsf{Unforg}}$, as otherwise we cannot win Π_{MDVS}'s unforgeability game; on the other hand we are also assuming there was no query to $\mathcal{O}_E$ that output the same verification key $\mathtt{vk}$. Since the Π_{DSS} verification key is part of the messages that are signed and/or verified using Π_{MDVS}, then if σ verifies as being a valid signature on $(\vec{v_\beta}_{\text{PKEBC}}, m, \mathtt{vk})$ with respect to sender public key $\mathtt{spk}_{i,\beta}$ and vector of verifier public keys $\vec{v_\beta}_{\text{MDVS}}$ using B_j's secret verification key $\mathtt{rsk}_j.\mathtt{vsk}$, the reduction wins the Unforg game of Π_{MDVS} (see Definition 25). Finally, if $\mathbf{A}$ only queries for at most $n_S \leq n_{S\,\text{MDVS}}$ (resp. $n_R \leq n_{V\,\text{MDVS}}$) different sender keys (resp. different receiver keys), makes up to $q_E \leq q_{S\,\text{MDVS}}$ queries to $\mathcal{O}_E$ and up to $q_D \leq q_{V\,\text{MDVS}}$ queries to $\mathcal{O}_D$, and the sum of lengths of the party vectors input to $\mathcal{O}_E$ is at most $d_E \leq d_{S\,\text{MDVS}}$, since from Equation 6.10 no adversary $(\varepsilon_{\text{MDVS-}\mathsf{Unforg}}, t_{\text{MDVS}})$-breaks the

$$(n_{S\,\text{MDVS}}, n_{V\,\text{MDVS}}, d_{S\,\text{MDVS}}, q_{S\,\text{MDVS}}, q_{V\,\text{MDVS}})\text{-Unforgeability}$$

of Π_{MDVS}, it follows

$$\left|\Pr[\mathbf{AG}^4_\beta = \mathtt{win}] - \Pr[\mathbf{AG}^3_\beta = \mathtt{win}]\right| \leq \varepsilon_{\text{MDVS-}\mathsf{Unforg}}.$$

$\mathbf{G}^4_\beta \rightsquigarrow \mathbf{G}^5_\beta$: The only difference between these games is that in $\mathbf{G}^5_\beta$ the distribution of PKEBC key-pairs returns to being the distribution induced by Π_{PKEBC}'s G algorithm. (This step is necessary to allow us to reduce to the $\mathsf{IND\text{-}CCA\text{-}2}$ and $\mathsf{IK\text{-}CCA\text{-}2}$ security of Π_{PKEBC}.) It then follows

$$\left|\Pr[\mathbf{AG}^5_\beta = \mathtt{win}] - \Pr[\mathbf{AG}^4_\beta = \mathtt{win}]\right| \leq \varepsilon_{\text{PKEBC-}\mathsf{Corr}}.$$

At this point we are finally set to make the security reductions to the $\mathsf{IND\text{-}CCA\text{-}2}$ and $\mathsf{IK\text{-}CCA\text{-}2}$ security of the underlying Π_{PKE}. For convenience, we now introduce one last hybrid, intermediate between $\mathbf{G}^5_{\mathbf{0}}$ and $\mathbf{G}^5_{\mathbf{1}}$.

$\mathbf{G}^5_{\mathbf{0}} \rightsquigarrow \mathbf{G}^6$: The only difference between $\mathbf{G}^5_{\mathbf{0}}$ and $\mathbf{G}^6$ is that in $\mathbf{G}^6$, the PKEBC ciphertext of each challenge query $\mathcal{O}_E\big((A_{i,\mathbf{0}}, \vec{V_{\mathbf{0}}}), (A_{i,\mathbf{1}}, \vec{V_{\mathbf{1}}}), m\big)$ is now an encryption of $(\mathtt{spk}_{i,\mathbf{1}}, \vec{v_{\mathbf{1}}}_{\text{MDVS}}, m, \sigma)$ under $\vec{v_{\mathbf{0}}}_{\text{PKEBC}}$, where $\mathtt{spk}_{i,\mathbf{1}}$ is $A_{i,\mathbf{1}}$'s public key,

$$\vec{v_{\mathbf{1}}}_{\text{MDVS}} \coloneqq (\mathtt{vpk}_{\text{MDVS}1,\mathbf{1}}, \ldots, \mathtt{vpk}_{\text{MDVS}|\vec{v}|,\mathbf{1}}), \text{ and}$$
$$\vec{v_{\mathbf{1}}}_{\text{PKEBC}} \coloneqq (\mathtt{pk}_{\text{PKEBC}1,\mathbf{1}}, \ldots, \mathtt{pk}_{\text{PKEBC}|\vec{v}|,\mathbf{1}})$$

are, respectively, the vectors of public MDVS verifier keys and public PKEBC receiver keys corresponding to $\vec{V_{\mathbf{1}}}$, and where

$$\sigma \leftarrow \Pi_{\text{MDVS}}.Sig_{\mathtt{pp}_{\text{MDVS}}}(\mathtt{ssk}_{\text{MDVS}i,\mathbf{1}}, \vec{v_{\mathbf{1}}}_{\text{MDVS}}, (\vec{v_{\mathbf{1}}}_{\text{PKEBC}}, m, \mathtt{vk})),$$

is an MDVS signature on $(\vec{v_{\mathbf{1}}}_{\text{PKEBC}}, m, \mathtt{vk})$, with $\mathtt{ssk}_{\text{MDVS}i,\mathbf{1}}$ being $A_{i,\mathbf{1}}$'s secret MDVS signing key and $\mathtt{vk}$ being the DSS verification key in c. So, if $\mathbf{A}$ only queries

for at most $n_R \leq n_{\text{PKEBC}}$ different receivers, the sum of lengths of the vectors input to $\mathcal{O}_E$ is at most $d_E \leq d_{E\text{PKEBC}}$, and $\mathbf{A}$ makes at most $q_E \leq q_{E\text{PKEBC}}$ and $q_D \leq q_{D\text{PKEBC}}$ queries to oracles $\mathcal{O}_E$ and $\mathcal{O}_D$, respectively, since from Equation 6.9 that no adversary $(t_{\text{PKEBC}}, \varepsilon_{\text{PKEBC-IND-CCA-2}})$-breaks the

$$(n_{\text{PKEBC}}, d_{E\text{PKEBC}}, q_{E\text{PKEBC}}, q_{D\text{PKEBC}})\text{-IND-CCA-2 security}$$

of Π_{PKEBC}, it follows

$$\left|\Pr[\mathbf{AG}^6 = \mathtt{win}] - \Pr[\mathbf{AG}_{\mathbf{0}}^4 = \mathtt{win}]\right| \leq \varepsilon_{\text{PKEBC-IND-CCA-2}}.$$

In a similar manner, we can now reduce an adversary distinguishing $\mathbf{G}^6$ and $\mathbf{G}_{\mathbf{1}}^5$ to one distinguishing the $\mathbf{G}_{\mathbf{0}}^{\mathsf{IK\text{-}CCA\text{-}2}}$ and $\mathbf{G}_{\mathbf{1}}^{\mathsf{IK\text{-}CCA\text{-}2}}$ games of the underlying Π_{PKEBC} scheme because we can handle all decryption queries—either by using the $\mathcal{O}_D$ oracle of Π_{PKEBC}'s IK-CCA-2 games, or by outputting $\perp$. It follows

$$\left|\Pr[\mathbf{AG}^6 = \mathtt{win}] - \Pr[\mathbf{AG}_{\mathbf{1}}^5 = \mathtt{win}]\right| \leq \varepsilon_{\text{PKEBC-IK-CCA-2}}.$$

I.6 Proof of Theorem 11

This proof proceeds via a sequence of games. In the following, let $\beta \in \{0, 1\}$.

$\mathbf{G}_\beta^{\mathsf{OTR}} \rightsquigarrow \mathbf{G}_\beta^1$: This game hop is analogous to $\mathbf{G}^{\mathsf{Unforg}} \rightsquigarrow \mathbf{G}^1$ from the proof of Theorem 8. It then follows

$$\left|\Pr[\mathbf{AG}_\beta^1 = \mathtt{win}] - \Pr[\mathbf{AG}_\beta^{\mathsf{OTR}} = \mathtt{win}]\right| \leq \varepsilon_{\text{DSS-1-sEUF-CMA}}.$$

By an argument similar to the one given in the proof of Theorem 8 one can see that at this point we can directly reduce an adversary distinguishing $\mathbf{G}_{\mathbf{0}}^1$ and $\mathbf{G}_{\mathbf{1}}^1$ to one distinguishing the $\mathbf{G}_{\mathbf{0}}^{\mathsf{OTR}}$ and $\mathbf{G}_{\mathbf{1}}^{\mathsf{OTR}}$ games of the underlying Π_{MDVS} scheme. (The main idea behind the argument is that one can use the $\mathcal{O}_V$ oracle of Π_{MDVS}'s underlying OTR game for handling decryption queries where the verification key $\mathtt{vk}$ was never output as part of a ciphertext by a query to $\mathcal{O}_E$; see Definition 26.) If $\mathbf{A}$ only queries for at most $n_S \leq n_{S\text{MDVS}}$ (resp. $n_R \leq n_{V\text{MDVS}}$) different sender keys (resp. different receiver keys), makes up to $q_E \leq q_{S\text{MDVS}}$ queries to $\mathcal{O}_E$ and up to $q_D \leq q_{V\text{MDVS}}$ queries to $\mathcal{O}_D$, and the sum of lengths of the party vectors input to $\mathcal{O}_E$ is at most $d_E \leq d_{S\text{MDVS}}$, since from Equation 6.15 no adversary $(\varepsilon_{\text{MDVS-OTR}}, t_{\text{MDVS}})$-breaks the

$$(n_{S\text{MDVS}}, n_{V\text{MDVS}}, d_{S\text{MDVS}}, q_{S\text{MDVS}}, q_{V\text{MDVS}})\text{-Off-The-Record security}$$

of Π_{MDVS}, it follows

$$\left|\Pr[\mathbf{AG}_{\mathbf{0}}^1 = \mathtt{win}] - \Pr[\mathbf{AG}_{\mathbf{1}}^1 = \mathtt{win}]\right| \leq \varepsilon_{\text{MDVS-OTR}}.$$

□