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).

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 [12,16,19–21,23,29–31], 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) [12].[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, 22]. 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 [22].

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) [13] which additionally gives consistency guarantees 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;

[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.

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

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 [26] 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; 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.[5]

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

[5] 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.

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, and then using the PKEBC scheme to encrypt the signed messages, together with their MDVS signatures.

Since an MDRS-PKE scheme is an extension of an MDVS scheme with privacy of identities and confidentiality, 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 [12].

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*) [10, 12]: 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[6], 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 [11] provide is *Forward Secrecy* [17]. 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. 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 [12], including the one given in this paper, do not capture this.

A natural generalization of two party secure messaging is secure group messaging [2, 12]. 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 [12]. 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 [12].

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 [12].

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

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 [12]. 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 [9, 13]. 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. Diament et al. introduce a special type of BE scheme, called Dual-Receiver Encryption schemes, which allow a sender to send messages to two (and only two) receivers. By limiting the number of receivers to two receivers, these schemes allow for efficient constructions with relatively short ciphertexts and public keys from standard assumptions.

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 [22]. 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 [12]. 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 $\vec{V} := (\text{Bob}, \text{Charlie})$, $\vec{v} := (\mathtt{pk}_{\text{Bob}}, \mathtt{pk}_{\text{Charlie}})$ is $\vec{V}$'s corresponding vector of public keys. In particular, V_1 is Bob and v_1 is Bob's public key $\mathtt{pk}_{\text{Bob}}$, and V_2 is Charlie and v_2 is Charlie's public key $\mathtt{pk}_{\text{Charlie}}$. 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 $\mathsf{IND\text{-}CCA\text{-}2}$ and $\mathsf{IK\text{-}CCA\text{-}2}$ 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 $c \leftarrow E_{\mathsf{pp},\vec{v}}(m)$.

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 $\mathtt{sk}_j$, $(\vec{v}, m) \leftarrow D_{\mathsf{pp},\mathsf{sk}_j}(c)$, 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.*

We say that an adversary $\mathbf{A}$ $(\varepsilon_{\mathsf{Corr}}, t)$-breaks the (n, d_E, q_E, q_D)-Correctness of a PKEBC scheme Π if $\mathbf{A}$ runs in time at most t, queries $\mathcal{O}_{PK}$, $\mathcal{O}_E$ and $\mathcal{O}_D$ on at most n different parties[7], 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 satisfies $Adv^{\mathsf{Corr}}(\mathbf{A}) \geq \varepsilon_{\mathsf{Corr}}$.

The following notion captures the guarantee that if a ciphertext c is an honestly generated ciphertext for a vector of receivers $\vec{R}$ (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 $B \notin \vec{R}$ 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* $Adv^{\mathsf{Rob}}(\mathbf{A})$*.*

[7] 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.

An adversary $\mathbf{A}$ $(\varepsilon_{\mathsf{Rob}}, t)$-breaks the Robustness of a PKEBC scheme Π if $\mathbf{A}$ runs in time at most t and satisfies $Adv^{\mathsf{Rob}}(\mathbf{A}) \geq \varepsilon_{\mathsf{Rob}}$.

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 [18]); 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, as we will see, if the PKE scheme underlying our PKEBC scheme's construction satisfies Correctness against unbounded adversaries, then the PKEBC scheme's construction can be proven to satisfy such stronger Correctness and Robustness security 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 $\vec{R}$ 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.*

We say that an adversary $\mathbf{A}$ $(\varepsilon_{\mathsf{Cons}}, t)$-breaks the (n, q_D)-Consistency of Π if $\mathbf{A}$ runs in time at most t, queries $\mathcal{O}_{SK}$, $\mathcal{O}_{PK}$ and $\mathcal{O}_D$ on at most n different parties, makes at most q_D queries to $\mathcal{O}_D$ and satisfies $Adv^{\mathsf{Cons}}(\mathbf{A}) \geq \varepsilon_{\mathsf{Cons}}$.

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 [24] that an analogous Consistency notion for MDVS schemes (introduced in [12]) implies composable security. Yet, as we will see, if the PKE scheme underlying our PKEBC scheme's construction satisfies Correctness against unbounded adversaries, then our PKEBC scheme can be proven to satisfy a stronger Consistency notion in which the adversary can query for any party's secret key.

The two following security notions are the multi-receiver variants of IND-CCA-2 security (introduced in [27]) 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 `test`;
2. Otherwise, compute and output $(\vec{v}, m) \leftarrow D_{\mathsf{pp},\mathtt{sk}_j}(c)$, 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*

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

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*

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

We say that an adversary $\mathbf{A}$ $(\varepsilon_{\mathsf{IND\text{-}CCA\text{-}2}}, t)$-breaks (resp. $(\varepsilon_{\mathsf{IK\text{-}CCA\text{-}2}}, t)$-breaks) the (n, d_E, q_E, q_D)-$\mathsf{IND\text{-}CCA\text{-}2}$ (resp. (n, d_E, q_E, q_D)-$\mathsf{IK\text{-}CCA\text{-}2}$) security of 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 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 $Adv^{\mathsf{IND\text{-}CCA\text{-}2}}(\mathbf{A}) \geq \varepsilon_{\mathsf{IND\text{-}CCA\text{-}2}}$ (resp. $Adv^{\mathsf{IK\text{-}CCA\text{-}2}}(\mathbf{A}) \geq \varepsilon_{\mathsf{IK\text{-}CCA\text{-}2}}$).

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;
- $(\varepsilon_{\mathsf{Rob}}, t)$-breaks the Robustness of Π;
- $(\varepsilon_{\mathsf{Cons}}, t)$-breaks the (n, q_D)-Consistency of Π;
- $(\varepsilon_{\mathsf{IND\text{-}CCA\text{-}2}}, t)$-breaks the (n, d_E, q_E, q_D)-$\mathsf{IND\text{-}CCA\text{-}2}$ security of Π; or
- $(\varepsilon_{\mathsf{IK\text{-}CCA\text{-}2}}, t)$-breaks the (n, d_E, q_E, q_D)-$\mathsf{IK\text{-}CCA\text{-}2}$ 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 [26] 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, 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 `pk` and a sequence of random coins r such that c is an encryption of m under `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].[8] Furthermore, we could use any perfectly correct PKE scheme as the statistically binding commitment scheme needed by our scheme (in particular ElGamal), and the tightly unbounded simulation sound NIZK scheme from [14].

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

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

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

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

4.1 Security Analysis of PKEBC Construction

We now prove the security of our PKEBC scheme construction. Refer to [25] for a full proof of the following results.

Theorem 1. *If Π_{PKE} is*

$$\begin{aligned} (\varepsilon_{\text{PKE-}\mathsf{Corr}}, \varepsilon_{\text{PKE-}\mathsf{IND\text{-}CPA}}, \varepsilon_{\text{PKE-}\mathsf{IK\text{-}CPA}}, \\ t_{\text{PKE}}, n_{\text{PKE}}, q_{E_{\text{PKE}}}, q_{D_{\text{PKE}}}, \mathsf{Corr})\text{-}\mathit{secure}, \end{aligned} \tag{4.3}$$

[8] In the full version of this paper, we show that ElGamal is also tightly multi-user multi-challenge $\mathsf{IK\text{-}CPA}$ secure under the DDH assumption (see [25]).

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

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

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

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

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

Π_{NIZK} *is*

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

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*

$$(n := n_{\mathrm{PKE}}, d_E := q_{E\,\mathrm{PKE}}, q_E := q_{P\,\mathrm{NIZK}}, q_D := \min(q_{V\,\mathrm{NIZK}}, q_{D\,\mathrm{PKE}}))\text{-Correctness,}$$

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

Proof Sketch. Algorithm 1 is composed of a CS, a NIZK and a PKE scheme. The correctness error of this PKEBC protocol is essentially the sum of the errors of these underlying schemes. We prove this by game hoping: we replace each scheme with a perfect version of itself, until the final game has correctness error 0. The advantage in distinguishing between the first and the last game is then the sum of advantages in distinguishing between the underlying schemes and the corresponding perfect versions. $\square$

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 correctness against unbounded adversaries, 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_{CS} is*

$$(\varepsilon_{\mathsf{CS\text{-}Hiding}}, \varepsilon_{\mathsf{CS\text{-}Binding}}, t_{\mathrm{CS}}, q_{\mathrm{CS}}, \mathsf{Binding})\text{-}secure, \tag{4.6}$$

then no adversary $\mathbf{A}$ (ε)-breaks Π's Robustness, with $\varepsilon > \varepsilon_{\mathsf{CS\text{-}Binding}}$.

Proof Sketch. To violate robustness, the same ciphertext must be the encryption of two plaintexts that have different vectors of receivers. But since a commitment to this vector (along with the message) is part of the ciphertext, there must be two vectors of receivers (and messages) that produce the same commitment. And the probability of this happening is bounded by $\varepsilon_{\mathsf{CS\text{-}Binding}}$. □

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 [14] has deterministic proof verification and is tightly unbounded simulation sound. The reason for this assumptions 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*

$$\begin{aligned}(\varepsilon_{\mathsf{PKE\text{-}Corr}}, \varepsilon_{\mathsf{PKE\text{-}IND\text{-}CPA}}, \varepsilon_{\mathsf{PKE\text{-}IK\text{-}CPA}},&\\ t_{\mathrm{PKE}}, n_{\mathrm{PKE}}, q_{E\,\mathrm{PKE}}, q_{D\,\mathrm{PKE}}, \mathsf{Corr})\text{-}secure,&\end{aligned} \tag{4.7}$$

Π_{NIZK} is

$$\begin{aligned}(\varepsilon_{\mathsf{NIZK\text{-}Complete}}, \varepsilon_{\mathsf{NIZK\text{-}Sound}}, \varepsilon_{\mathsf{NIZK\text{-}ZK}}, \varepsilon_{\mathsf{NIZK\text{-}SS}},&\\ t_{\mathrm{NIZK}}, q_{P\,\mathrm{NIZK}}, q_{V\,\mathrm{NIZK}})\text{-}secure,&\end{aligned} \tag{4.8}$$

Π_{CS} is

$$(\varepsilon_{\mathsf{CS\text{-}Hiding}}, \varepsilon_{\mathsf{CS\text{-}Binding}}, t_{\mathrm{CS}}, q_{\mathrm{CS}}, \mathsf{Binding})\text{-}secure, \tag{4.9}$$

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

$$(n := n_{\mathrm{PKE}}, q_D := q_{V\,\mathrm{NIZK}})\text{-}Consistency,$$

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

Proof Sketch. As in the proof of Theorem 1, we proceed by game hoping and replace the CS and NIZK by ideal versions. We then show that if the underlying PKE has perfect correctness, consistency cannot be violated. Hence the final error is that of the CS, the soundness of the NIZK and the correctness of the PKE. □

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 something stronger, namely that Π is Consistent with respect to a stronger Consistency notion which allows adversaries to query for the secret key of any honest receiver.

Theorem 4. *If Π_{PKE} is*

$$\begin{aligned}(&\varepsilon_{\mathrm{PKE\text{-}\mathsf{Corr}}}, \varepsilon_{\mathrm{PKE\text{-}\mathsf{IND\text{-}CPA}}}, \varepsilon_{\mathrm{PKE\text{-}\mathsf{IK\text{-}CPA}}}, \\ &t_{\mathrm{PKE}}, n_{\mathrm{PKE}}, q_{E\,\mathrm{PKE}}, q_{D\,\mathrm{PKE}}, \mathsf{Corr})\text{-}secure,\end{aligned} \tag{4.10}$$

Π_{NIZK} is

$$\begin{aligned}(&\varepsilon_{\mathrm{NIZK\text{-}\mathsf{Complete}}}, \varepsilon_{\mathrm{NIZK\text{-}\mathsf{Sound}}}, \varepsilon_{\mathrm{NIZK\text{-}\mathsf{ZK}}}, \varepsilon_{\mathrm{NIZK\text{-}\mathsf{SS}}}, \\ &t_{\mathrm{NIZK}}, q_{P\,\mathrm{NIZK}}, q_{V\,\mathrm{NIZK}})\text{-}secure,\end{aligned} \tag{4.11}$$

and Π_{CS} is

$$(\varepsilon_{\mathrm{CS\text{-}\mathsf{Hiding}}}, \varepsilon_{\mathrm{CS\text{-}\mathsf{Binding}}}, t_{\mathrm{CS}}, q_{\mathrm{CS}}, \mathsf{Binding})\text{-}secure, \tag{4.12}$$

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

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

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}.

Proof Sketch. The definition of IK-CCA-2 security bounds the ability of the adversary to distinguish between two games, one of which generates challenge ciphertexts encrypted for the vector of receivers $\vec{V}_0$ and the other for $\vec{V}_1$. The full proof is a simple generalization of the one from [28] and consists of 16 game hops that bound an adversary's advantage in distinguishing the two game systems. Here we highlight the main ideas in this proof.

The first step in the proof is replacing the NIZK proofs with simulated ones, as this allows creating valid NIZK proofs for false statements. Recall that each PKEBC ciphertext includes, for each receiver, two encryptions of the same plaintext under the two different (and independent) public keys of the receiver. This allows being able to answer the adversary's decryption queries while only knowing one of the two secret keys of the receiver, which is crucial for the reductions to the IND-CPA and IK-CPA security for the underlying PKE scheme. Another key step in the proof is relying on the Simulation Soundness of the underlying NIZK scheme to be able to change the key used for answering decryption queries. Finally, the last main technical idea in the proof is making the sequence of random coins ρ encrypted using the underlying PKE scheme independent of the random coins actually used by the underlying Commitment Scheme when reducing to its Hiding property. $\square$

Theorem 5. *If Π_{PKE} is*

$$\begin{aligned}(&\varepsilon_{\text{PKE-}\mathsf{Corr}}, \varepsilon_{\text{PKE-}\mathsf{IND\text{-}CPA}}, \varepsilon_{\text{PKE-}\mathsf{IK\text{-}CPA}}, \\ &\qquad t_{\text{PKE}}, n_{\text{PKE}}, q_{E_{\text{PKE}}}, q_{D_{\text{PKE}}}, \mathsf{Corr})\text{-}\textit{secure},\end{aligned} \tag{4.13}$$

Π_{NIZK} is

$$\begin{aligned}(&\varepsilon_{\text{NIZK-}\mathsf{Complete}}, \varepsilon_{\text{NIZK-}\mathsf{Sound}}, \varepsilon_{\text{NIZK-}\mathsf{ZK}}, \varepsilon_{\text{NIZK-}\mathsf{SS}}, \\ &\qquad t_{\text{NIZK}}, q_{P_{\text{NIZK}}}, q_{V_{\text{NIZK}}})\text{-}\textit{secure},\end{aligned} \tag{4.14}$$

and Π_{CS} is

$$(\varepsilon_{\text{CS-}\mathsf{Hiding}}, \varepsilon_{\text{CS-}\mathsf{Binding}}, t_{\text{CS}}, q_{\text{CS}}, \mathsf{Binding})\text{-}\textit{secure}, \tag{4.15}$$

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

$$\begin{aligned}&(n \coloneqq n_{\text{PKE}}, d_E \coloneqq q_{E_{\text{PKE}}}, \\ &q_E \coloneqq \min(q_{P_{\text{NIZK}}}, q_{\text{CS}}), q_D \coloneqq q_{V_{\text{NIZK}}})\text{-}\mathsf{IND\text{-}CCA\text{-}2}\ \textit{security},\end{aligned}$$

with

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

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}.

Proof Sketch. The proof of this theorem is a simple adaptation of the proof of Theorem 4, but where one no longer makes game hopping on the IK-CPA security of the PKE scheme Π_{PKE} underlying PKEBC scheme Π's construction. $\square$

5 Multi-Designated Receiver Signed Public Key Encryption Schemes

We now introduce the second new type of scheme we give in this paper: Multi-Designated Receiver Signed Public Key Encryption (MDRS-PKE). An MDRS-PKE scheme $\Pi = (S, G_S, G_V, E, D)$ with message space $\mathcal{M}$ is a five-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_V: 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.

5.1 The Security of MDRS-PKE Schemes

Below we state the definitions of Correctness, Consistency, Unforgeability, $\mathsf{IND\text{-}CCA\text{-}2}$ security, $\mathsf{IK\text{-}CCA\text{-}2}$ 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 $\Pi = (S, G_S, G_V, E, D)$ 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. $(\mathtt{vpk}_j, \mathtt{vsk}_j) \leftarrow \mathcal{O}_{RK}(B_j)$;

2. Output $(\mathtt{spk}, \vec{v} \coloneqq (\mathtt{rpk}_1, \ldots, \mathtt{rpk}_{|\vec{v}|}), m) \leftarrow D_{\mathsf{pp}}(\mathtt{vsk}_j, c)$.

We now introduce the game-based notions. Let $\Pi = (S, G_S, G_V, E, D)$ 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 are two queries* q_E *and* q_D *to* $\mathcal{O}_E$ *and* $\mathcal{O}_D$*, respectively, where* q_E *has input* $(A_i, \vec{V}, m)$ *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* $(\mathtt{spk}_i', \vec{v}', m')$ *with* $(\mathtt{spk}_i', \vec{v}', m') = \perp$ *or* $(\mathtt{spk}_i', \vec{v}', m') \neq (\mathtt{spk}_i, \vec{v}, m)$*—where* $\mathtt{spk}_i$ *is* A_i*'s public key and* $\vec{v}$ *is the corresponding vector of public keys of the parties of* $\vec{V}$*— 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 [24]—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 [24]). As noted in Remark 9, 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 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* $\mathcal{O}_{RK}$*, query* $\mathcal{O}_D(B_i, c)$ *outputs some* $(\mathtt{spk}_l, \vec{v}, m)$ *satisfying* $(\mathtt{spk}_l, \vec{v}, m) \neq \perp$*,* $\mathtt{spk}_l$ *is some party* A_l*'s public sender key (i.e.* $\mathcal{O}_{SPK}(A_l) = \mathtt{spk}_l$*) and* $\mathtt{rpk}_j \in \vec{v}$ *(where* $\mathtt{rpk}_j$ *is* B_j*'s public key), and query* $\mathcal{O}_D(B_j, c)$ *does not output the same triple* $(\mathtt{spk}_l, \vec{v}, m)$*.*

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.*

The following security notion is analogous to the $\mathsf{EUF\text{-}CMA}$ security notion for Digital Signature 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 it some message that A actually did not send.

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

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

We say that $\mathbf{A}$ *wins the game if there is a query* q *to* $\mathcal{O}_D$ *on an input* (B_j, c) *that outputs* $(\mathtt{spk}_i, \vec{v}, m) \neq \perp$ *with* $\mathtt{spk}_i$ *being some party* A_i*'s sender public key (i.e.* $\mathcal{O}_{SPK}(A_i) = \mathtt{spk}_i$*), there was no query* $\mathcal{O}_E(A_i, \vec{V}, m)$ *where* $\vec{V}$ *is the vector of parties with corresponding public keys* $\vec{v}$*,* $\mathcal{O}_{SK}$ *was not queried on input* A_i*, and* $\mathcal{O}_{RK}$ *was not queried on input* 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* $Adv^{\mathsf{Unforg}}(\mathbf{A})$*.*

We say that an adversary $\mathbf{A}$ (ε, t)-breaks the $(n_S, n_R, d_E, q_E, q_D)$-Correctness, Consistency, or Unforgeability 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.

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:

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 $(\mathtt{spk}_i, \vec{v}, m) \leftarrow D_{\mathsf{pp},\mathtt{sk}_j}(c)$, where $\mathtt{sk}_j$ is B_j's secret key; output $(\mathtt{spk}_i, \vec{v}, m)$.

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 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* $\mathbf{b} \in \{0, 1\}$*:*

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

$\mathbf{A}$ *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*

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

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 **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* $\mathbf{b} \in \{0,1\}$*:*

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

$\mathbf{A}$ *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*

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

We say that an adversary $\mathbf{A}$ (ε, t)-breaks the (n_R, d_E, q_E, q_D)-IND-CCA-2 security or IK-CCA-2 security of Π if $\mathbf{A}$ runs in time at most t, queries $\mathcal{O}_{RPK}$, $\mathcal{O}_E$ 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 has at least ε advantage in winning the corresponding security game.

Remark 6. The IND-CCA-2 and IK-CCA-2 security 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, 15] 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.

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 [12] for MDVS schemes. This notion defines two game systems, $\mathbf{G}_{\mathbf{0}}^{\mathsf{OTR}\text{-}Forge}$ and $\mathbf{G}_{\mathbf{1}}^{\mathsf{OTR}\text{-}Forge}$, which are parameterized by an algorithm *Forge*. The game systems also provide adversaries with access to an oracle $\mathcal{O}_E$, whose behavior varies depending on the underlying game system, i.e. depending on $\mathbf{b} \in \{0,1\}$. $\mathcal{O}_E$ behaves as follows:

Encryption Oracle: $\mathcal{O}_E(\mathtt{type} \in \{\mathtt{sign}, \mathtt{forge}\}, A_i, \vec{V}, m, \mathcal{D})$

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

1. $c_{\mathbf{0}} \leftarrow E_{\mathsf{pp}}(\mathtt{ssk}_i, \vec{v}, m)$;
2. $c_{\mathbf{1}} \leftarrow \mathit{Forge}_{\mathsf{pp}}(\mathtt{spk}_i, \vec{v}, m, \{\mathtt{rsk}_j\}_{B_j \in \mathcal{D}})$;
3. If $\mathbf{b} = 0$, output $c_{\mathbf{0}}$ if $\mathtt{type} = \mathtt{sign}$ and $c_{\mathbf{1}}$ if $\mathtt{type} = \mathtt{forge}$;
4. Otherwise, if $\mathbf{b} = 1$, output $c_{\mathbf{1}}$.

Definition 11 (Off-The-Record). *Let Forge be a PPT algorithm that on input* $\mathtt{pp}$, $\mathtt{spk}_{i^*}$, $\vec{v}$, m^* *and* $\{\mathtt{rsk}_j\}_{B_j \in \mathcal{D}^*}$*, outputs a forged ciphertext* c'*. For* $\mathbf{b} \in \{0, 1\}$*, consider the following game played between an adversary* $\mathbf{A}$ *and game system* $\mathbf{G}_{\mathbf{b}}^{\mathsf{OTR}\text{-}\mathit{Forge}}$*:*

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

$\mathbf{A}$ *wins the game if* $b' = \mathbf{b}$ *and for every query* $(\mathtt{type}, A_i, \vec{V}, m, \mathcal{D})$ *to* $\mathcal{O}_E$*, and letting* c *be the output of* $\mathcal{O}_E$*, all of the following hold:*

1. $\mathcal{D} \subseteq \mathit{Set}(\vec{V})$*;*
2. *for every query* B_j *to* $\mathcal{O}_{VK}$*,* $B_j \notin \mathit{Set}(\vec{V}) \setminus \mathcal{D}$*;*
3. *for every query* A_l *to* $\mathcal{O}_{SK}$*,* $A_l \neq A_i$*; and*
4. *for all queries* $\mathcal{O}_D(A_l, B_j, \vec{V}', m', c')$ *with* $A_l = A_i$ *and* $\vec{V}' = \vec{V}$*,* $c' \neq c$*.*

$\mathbf{A}$*'s advantage in winning the Off-The-Record security game with respect to Forge is defined as*

$$\mathit{Adv}^{\mathsf{OTR}\text{-}\mathit{Forge}}(\mathbf{A}) := \left| \Pr[\mathbf{A}\mathbf{G}_{\mathbf{0}}^{\mathsf{OTR}\text{-}\mathit{Forge}} = \mathtt{win}] + \Pr[\mathbf{A}\mathbf{G}_{\mathbf{1}}^{\mathsf{OTR}\text{-}\mathit{Forge}} = \mathtt{win}] - 1 \right|.$$

We say that an adversary $\mathbf{A}$ $(\varepsilon_{\mathsf{OTR}}, t)$-breaks the $(n_S, n_R, d_E, q_E, q_D)$-Off-The-Record security of Π with respect to algorithm *Forge* 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 satisfies $\mathit{Adv}^{\mathsf{OTR}\text{-}\mathit{Forge}}(\mathbf{A}) \geq \varepsilon_{\mathsf{OTR}}$.

Finally, we say that Π is

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

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_R, d_E, q_E, q_D)-IND-CCA-2 security of Π;
– $(\varepsilon_{\mathsf{IK\text{-}CCA\text{-}2}}, t)$-breaks the (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 with respect to *Forge*.

Remark 7. 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 give a construction of an MDRS-PKE scheme from a PKEBC scheme and an MDVS scheme (see Algorithm 2). The construction essentially consists of using the MDVS scheme to sign both the messages and the vectors of public PKEBC keys of the receivers, and then using the PKEBC scheme to encrypt the signed message, together with its MDVS signature, the public MDVS signer key of the sender and the vector of public MDVS verifier keys of the receivers.

Remark 8. 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

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

$\mathit{Setup}(1^k)$
  $\mathtt{pp}_{\text{MDVS}} \leftarrow \Pi_{\text{MDVS}}.\mathit{Setup}(1^k)$
  $\mathtt{pp}_{\text{PKEBC}} \leftarrow \Pi_{\text{PKEBC}}.S(1^k)$
  $\mathtt{pp} := (\mathtt{pp}_{\text{MDVS}}, \mathtt{pp}_{\text{PKEBC}})$
  return $\mathtt{pp}$

$G_S(\mathtt{pp} := (\mathtt{pp}_{\text{MDVS}}, \mathtt{pp}_{\text{PKEBC}}))$
  $(\mathtt{spk}_{\text{MDVS}}, \mathtt{ssk}_{\text{MDVS}}) \leftarrow \Pi_{\text{MDVS}}.G_S(\mathtt{pp}_{\text{MDVS}})$
  $\mathtt{spk} := \mathtt{spk}_{\text{MDVS}}$
  $\mathtt{ssk} := (\mathtt{spk}, \mathtt{ssk}_{\text{MDVS}})$
  return $(\mathtt{spk}, \mathtt{ssk})$

$G_V(\mathtt{pp} := (\mathtt{pp}_{\text{MDVS}}, \mathtt{pp}_{\text{PKEBC}}))$
  $(\mathtt{vpk}_{\text{MDVS}}, \mathtt{vsk}_{\text{MDVS}}) \leftarrow \Pi_{\text{MDVS}}.G_V(\mathtt{pp}_{\text{MDVS}})$
  $(\mathtt{pk}_{\text{PKEBC}}, \mathtt{sk}_{\text{PKEBC}}) \leftarrow \Pi_{\text{PKEBC}}.G(\mathtt{pp}_{\text{PKEBC}})$
  $\mathtt{rpk} := (\mathtt{vpk}_{\text{MDVS}}, \mathtt{pk}_{\text{PKEBC}})$
  $\mathtt{rsk} := \big(\mathtt{rpk}, (\mathtt{vsk}_{\text{MDVS}}, \mathtt{sk}_{\text{PKEBC}})\big)$
  return $(\mathtt{rpk}, \mathtt{rsk})$

$E_{\mathtt{pp}}(\mathtt{ssk}_i, \vec{v}, m)$
  With
    $\mathtt{pp} := (\mathtt{pp}_{\text{MDVS}}, \mathtt{pp}_{\text{PKEBC}})$
    $\mathtt{ssk}_i := (\mathtt{spk}_i, \mathtt{ssk}_{\text{MDVS}\,i})$
    $\vec{v} := \big(\mathtt{rpk}_1, \ldots, \mathtt{rpk}_{|\vec{v}|}\big)$
    for each $i \in \{1, \ldots, |\vec{v}|\}$
      $\mathtt{rpk}_i := (\mathtt{vpk}_{\text{MDVS}\,i}, \mathtt{pk}_{\text{PKEBC}\,i})$

  $\vec{v}_{\text{PKEBC}} \leftarrow (\mathtt{pk}_{\text{PKEBC}\,1}, \ldots, \mathtt{pk}_{\text{PKEBC}\,|\vec{v}|})$
  $\vec{v}_{\text{MDVS}} \leftarrow (\mathtt{vpk}_{\text{MDVS}\,1}, \ldots, \mathtt{vpk}_{\text{MDVS}\,|\vec{v}|})$
  $\sigma \leftarrow \Pi_{\text{MDVS}}.\mathit{Sign}_{\mathtt{pp}_{\text{MDVS}}}(\mathtt{ssk}_{\text{MDVS}\,i}, \text{Set}(\vec{v}_{\text{MDVS}}), (\vec{v}_{\text{PKEBC}}, m))$
  return $\Pi_{\text{PKEBC}}.E_{\mathtt{pp}_{\text{PKEBC}}}\big(\vec{v}_{\text{PKEBC}}, (\mathtt{spk}_i, \vec{v}_{\text{MDVS}}, m, \sigma)\big)$

$D_{\mathtt{pp}}(\mathtt{rsk}_j, c)$
  With
    $\mathtt{pp} := (\mathtt{pp}_{\text{MDVS}}, \mathtt{pp}_{\text{PKEBC}})$
    $\mathtt{rsk}_j := \big(\mathtt{rpk}_j, (\mathtt{vsk}_{\text{MDVS}\,j}, \mathtt{sk}_{\text{PKEBC}\,j})\big)$
    $\mathtt{rpk}_j := (\mathtt{vpk}_{\text{MDVS}\,j}, \mathtt{pk}_{\text{PKEBC}\,j})$

  $\big(\vec{v}_{\text{PKEBC}}, (\mathtt{spk}_i, \vec{v}_{\text{MDVS}}, m, \sigma)\big) \leftarrow \Pi_{\text{PKEBC}}.D_{\mathtt{pp}_{\text{PKEBC}}}(\mathtt{sk}_{\text{PKEBC}\,j}, c)$
  if $\big(\vec{v}_{\text{PKEBC}}, (\mathtt{spk}_i, \vec{v}_{\text{MDVS}}, m, \sigma)\big) = \bot \quad \vee \quad |\vec{v}_{\text{PKEBC}}| \neq |\vec{v}_{\text{MDVS}}|$ **then**
    return $\bot$
  $\vec{v} := \big((v_{\text{MDVS}\,1}, v_{\text{PKEBC}\,1}), \ldots, (v_{\text{MDVS}\,|\vec{v}_{\text{PKEBC}}|}, v_{\text{PKEBC}\,|\vec{v}_{\text{PKEBC}}|})\big)$
  if $\mathtt{rpk}_j \notin \vec{v}$ **then**
    return $\bot$
  if $\Pi_{\text{MDVS}}.\mathit{Vfy}_{\mathtt{pp}_{\text{MDVS}}}(\mathtt{spk}_i, \mathtt{vsk}_{\text{MDVS}\,j}, \text{Set}(\vec{v}_{\text{MDVS}}), (\vec{v}_{\text{PKEBC}}, m), \sigma) \neq \mathtt{valid}$ **then**
    return $\bot$
  return $(\mathtt{spk}_i, \vec{v}, m)$

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 [24][9]. 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

The security of our MDRS-PKE scheme follows from the security of the underlying PKEBC and MDVS schemes. For a full proof of these results, refer to [25].

Theorem 6. *If Π_{PKEBC} is*

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

and Π_{MDVS} is

$$\begin{aligned}(&\varepsilon_{\text{MDVS-}\mathsf{Corr}}, \varepsilon_{\text{MDVS-}\mathsf{Cons}}, \varepsilon_{\text{MDVS-}\mathsf{Unforg}}, \varepsilon_{\text{MDVS-}\mathsf{OTR}}, \varepsilon_{\text{MDVS-}\mathsf{PI}},\\ &t_{\text{MDVS}}, n_{S\text{MDVS}}, n_{V\text{MDVS}}, d_{S\text{MDVS}},\\ &q_{S\text{MDVS}}, q_{V\text{MDVS}}, \mathit{Forge}_{\text{MDVS}})\text{-}secure,\end{aligned} \tag{6.2}$$

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

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

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

Proof Sketch. To prove this theorem one introduces an intermediate game that assumes the correctness of the Π_{PKEBC} scheme underlying Π's construction. Then, one shows that the advantage of any adversary in winning this intermediate game can only differ from the advantage in winning the original game by at most the advantage that an adversary could have in winning the Correctness game of Π_{PKEBC}. Finally, one shows that the advantage in winning the new game is bound by the advantage in winning the Correctness game of the MDVS scheme Π_{MDVS} underlying Π's construction. □

[9] The composable notions capturing the security of MDVS given in [24] actually assume something even stronger: every dishonest party has access to the secret keys of every other dishonest party.

Remark 9. 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*

$$\begin{aligned}(&\varepsilon_{\text{PKEBC-}\mathsf{Corr}}, \varepsilon_{\text{PKEBC-}\mathsf{Rob}}, \varepsilon_{\text{PKEBC-}\mathsf{Cons}}, \varepsilon_{\text{PKEBC-}\mathsf{IND\text{-}CCA\text{-}2}}, \varepsilon_{\text{PKEBC-}\mathsf{IK\text{-}CCA\text{-}2}},\\ &t_{\text{PKEBC}}, n_{\text{PKEBC}}, d_{E\text{PKEBC}}, q_{E\text{PKEBC}}, q_{D\text{PKEBC}})\text{-}secure,\end{aligned} \tag{6.3}$$

and Π_{MDVS} is

$$\begin{aligned}(&\varepsilon_{\text{MDVS-}\mathsf{Corr}}, \varepsilon_{\text{MDVS-}\mathsf{Cons}}, \varepsilon_{\text{MDVS-}\mathsf{Unforg}}, \varepsilon_{\text{MDVS-}\mathsf{OTR}}, \varepsilon_{\text{MDVS-}\mathsf{PI}},\\ &t_{\text{MDVS}}, n_{S\text{MDVS}}, n_{V\text{MDVS}}, d_{S\text{MDVS}},\\ &q_{S\text{MDVS}}, q_{V\text{MDVS}}, \mathit{Forge}_{\text{MDVS}})\text{-}secure,\end{aligned} \tag{6.4}$$

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

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

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

Proof Sketch. To win the Consistency game, an adversary has to make two queries $\mathcal{O}_D(B_i, c)$ and $\mathcal{O}_D(B_j, c)$ such that $\mathcal{O}_D(B_i, c)$ outputs some $(\mathtt{spk}_l, \vec{v}, m) \neq \perp$— where $\mathtt{spk}_l$ is some party A_l's public sender key and B_j's public key is in $\vec{v}$, query $\mathcal{O}_D(B_j, c)$ does not output the same as $\mathcal{O}_D(B_i, c)$, and there is no query to $\mathcal{O}_{RK}$ on B_i or B_j. Note that this is the only possible way to win the Consistency game. With this, one then shows that an adversary winning the Consistency implies that it either broke the consistency of the PKEBC scheme Π_{PKEBC} underlying Π's construction, or that it broke the consistency of the MDVS scheme Π_{MDVS} underlying Π's construction. $\square$

Theorem 8. *If Π_{MDVS} is*

$$\begin{aligned}(&\varepsilon_{\text{MDVS-}\mathsf{Corr}}, \varepsilon_{\text{MDVS-}\mathsf{Cons}}, \varepsilon_{\text{MDVS-}\mathsf{Unforg}}, \varepsilon_{\text{MDVS-}\mathsf{OTR}}, \varepsilon_{\text{MDVS-}\mathsf{PI}},\\ &t_{\text{MDVS}}, n_{S\text{MDVS}}, n_{V\text{MDVS}}, d_{S\text{MDVS}},\\ &q_{S\text{MDVS}}, q_{V\text{MDVS}}, \mathit{Forge}_{\text{MDVS}})\text{-}secure,\end{aligned} \tag{6.5}$$

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

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

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

Proof Sketch. Any adversary for Π's Unforgeability game can be easily reduced into an adversary for the Unforgeability game of the MDVS scheme Π_{MDVS} underlying Π's construction that has the same advantage in winning Π_{MDVS}'s Unforgeability game. $\square$

Theorem 9. *If Π_{PKEBC} is*

$$(\varepsilon_{\mathrm{PKEBC}\text{-}\mathsf{Corr}}, \varepsilon_{\mathrm{PKEBC}\text{-}\mathsf{Rob}}, \varepsilon_{\mathrm{PKEBC}\text{-}\mathsf{Cons}}, \varepsilon_{\mathrm{PKEBC}\text{-}\mathsf{IND\text{-}CCA\text{-}2}}, \varepsilon_{\mathrm{PKEBC}\text{-}\mathsf{IK\text{-}CCA\text{-}2}}, t_{\mathrm{PKEBC}}, n_{\mathrm{PKEBC}}, d_{E\mathrm{PKEBC}}, q_{E\mathrm{PKEBC}}, q_{D\mathrm{PKEBC}})\text{-}\mathit{secure}, \tag{6.6}$$

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

$$(n_R \coloneqq n_{\mathrm{PKEBC}}, d_E \coloneqq d_{E\mathrm{PKEBC}}, q_E \coloneqq q_{E\mathrm{PKEBC}}, q_D \coloneqq q_{D\mathrm{PKEBC}})\text{-}\mathsf{IND\text{-}CCA\text{-}2}\ \mathit{security},$$

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

Proof Sketch. Distinguishing Π's (MDRS-PKE) IND-CCA-2 security games can be trivially reduced to distinguishing Π_{PKEBC}'s (PKEBC) IND-CCA-2 security games with the same advantage. $\square$

Remark 10. 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 $\mathcal{O}_E$. Yet, the proofs of Theorems 9 and 10 actually show something stronger. Namely, that Π is secure according to even the stronger IND-CCA-2 and IK-CCA-2 security notions in which an adversary is allowed to query for the secret key of any sender.

Theorem 10. *If Π_{PKEBC} is*

$$(\varepsilon_{\mathrm{PKEBC}\text{-}\mathsf{Corr}}, \varepsilon_{\mathrm{PKEBC}\text{-}\mathsf{Rob}}, \varepsilon_{\mathrm{PKEBC}\text{-}\mathsf{Cons}}, \varepsilon_{\mathrm{PKEBC}\text{-}\mathsf{IND\text{-}CCA\text{-}2}}, \varepsilon_{\mathrm{PKEBC}\text{-}\mathsf{IK\text{-}CCA\text{-}2}}, t_{\mathrm{PKEBC}}, n_{\mathrm{PKEBC}}, d_{E\mathrm{PKEBC}}, q_{E\mathrm{PKEBC}}, q_{D\mathrm{PKEBC}})\text{-}\mathit{secure}, \tag{6.7}$$

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

$$(n_R \coloneqq n_{\mathrm{PKEBC}}, d_E \coloneqq d_{E\mathrm{PKEBC}}, q_E \coloneqq q_{E\mathrm{PKEBC}}, q_D \coloneqq q_{D\mathrm{PKEBC}})\text{-}\mathsf{IK\text{-}CCA\text{-}2}\ \mathit{security},$$

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

Proof Sketch. To prove this theorem we introduce an intermediate game which is just like $\mathbf{G}_{\mathbf{0}}^{\mathsf{IK\text{-}CCA\text{-}2}}$ except that the $\mathcal{O}_E$ oracle behaves slightly differently: for each query $((A_{i,0}, \vec{V}_0), (A_{i,1}, \vec{V}_1), m)$, $\mathcal{O}_E$ behaves exactly as $\mathbf{G}_{\mathbf{0}}^{\mathsf{IK\text{-}CCA\text{-}2}}$'s $\mathcal{O}_E$ oracle would behave, except that now it encrypts the PKEBC ciphertext using the vector of public PKEBC keys corresponding to $\vec{V}_1$, rather than using the vector of public keys corresponding to $\vec{V}_0$. An adversary trying to distinguishing $\mathbf{G}_{\mathbf{0}}^{\mathsf{IK\text{-}CCA\text{-}2}}$ from this intermediate game can be trivially reduced to an adversary distinguishing the two IK-CCA-2 game systems for the underlying Π_{PKEBC} with the same distinguishing advantage; an adversary distinguishing the intermediate game from $\mathbf{G}_1^{\textsf{IK-CCA-2}}$ can be (again trivially) reduced to an adversary distinguishing the two IND-CCA-2 game systems for the underlying Π_{PKEBC} with the same distinguishing advantage. □

Algorithm 3 *Forge* algorithm for the construction given in Algorithm 2. In the following, let Π_{MDVS} and Π_{PKEBC} respectively be the MDVS and PKEBC schemes underlying the construction given in Algorithm 2, $\mathit{Forge}_{\text{MDVS}}$ be a signature forging algorithm for Π_{MDVS}, and $\{\mathtt{rsk}_{j'}\}_{B_{j'}\in\mathcal{D}}$ be the set of secret receiver keys of $\mathcal{D}$, the set of dishonest parties.

$\mathit{Forge}_{\mathsf{pp}}(\mathsf{spk}_i, \vec{v}, m, \{\mathsf{rsk}_{j'}\}_{B_{j'}\in\mathcal{D}})$
With
$\mathsf{pp} \coloneqq (\mathsf{pp}_{\text{MDVS}}, \mathsf{pp}_{\text{PKEBC}})$
$\mathsf{spk}_i \coloneqq \mathsf{spk}_{\text{MDVS}\,i}$
for each $\mathsf{rsk}_j \in \{\mathsf{rsk}_{j'}\}_{B_{j'}\in\mathcal{D}}$
$\mathsf{rsk}_j \coloneqq \big((\mathsf{vpk}_{\text{MDVS}\,j}, \mathsf{pk}_{\text{PKEBC}\,j}), (\mathsf{vsk}_{\text{MDVS}\,j}, \mathsf{sk}_{\text{PKEBC}\,j})\big)$
$\vec{v} \coloneqq \big(\mathsf{rpk}_1, \ldots, \mathsf{rpk}_{|\vec{v}|}\big)$
for each $i \in \{1, \ldots, |\vec{v}|\}$
$\mathsf{rpk}_i = (\mathsf{vpk}_{\text{MDVS}\,i}, \mathsf{pk}_{\text{PKEBC}\,i})$

$\vec{v}_{\text{PKEBC}} \leftarrow (\mathsf{pk}_{\text{PKEBC}\,1}, \ldots, \mathsf{pk}_{\text{PKEBC}\,|\vec{v}|})$
$\vec{v}_{\text{MDVS}} \leftarrow (\mathsf{vpk}_{\text{MDVS}\,1}, \ldots, \mathsf{vpk}_{\text{MDVS}\,|\vec{v}|})$
$\sigma_{\text{MDVS}} \leftarrow \mathit{Forge}_{\text{MDVS}_{\mathsf{pp}_{\text{MDVS}}}}(\mathsf{spk}_{\text{MDVS}\,i}, \mathrm{Set}(\vec{v}_{\text{MDVS}}), (\vec{v}_{\text{PKEBC}}, m), \{\mathsf{vsk}_{\text{MDVS}\,j'}\}_{B_{j'}\in\mathcal{D}})$
return $\Pi_{\text{PKEBC}}.E_{\mathsf{pp}_{\text{PKEBC}}}\big(\vec{v}_{\text{PKEBC}}, (\mathsf{spk}_{\text{MDVS}\,i}, \vec{v}_{\text{MDVS}}, m, \sigma_{\text{MDVS}})\big)$

Theorem 11. *In the following let Forge denote Algorithm 3. If* Π_{MDVS} *is*

$$\begin{aligned}(&\varepsilon_{\text{MDVS-}\textsf{Corr}}, \varepsilon_{\text{MDVS-}\textsf{Cons}}, \varepsilon_{\text{MDVS-}\textsf{Unforg}}, \varepsilon_{\text{MDVS-}\textsf{OTR}}, \varepsilon_{\text{MDVS-}\textsf{PI}},\\ &t_{\text{MDVS}}, n_{S\,\text{MDVS}}, n_{V\,\text{MDVS}}, d_{S\,\text{MDVS}},\\ &q_{S\,\text{MDVS}}, q_{V\,\text{MDVS}}, \mathit{Forge}_{\text{MDVS}})\text{-secure,}\end{aligned} \tag{6.8}$$

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

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

with $\varepsilon > \varepsilon_{\text{MDVS-}\textsf{OTR}}$*, and* $t_{\text{MDVS}} \approx t + t_{\textsf{OTR}}$*, where* $t_{\textsf{OTR}}$ *is the time to run* Π*'s* $\mathbf{G}^{\textsf{OTR}}$ *games.*

Proof Sketch. Any adversary for Π's Off-The-Record games can trivially be reduced into an adversary for the Off-The-Record games of the Π_{MDVS} scheme underlying Π's construction that has the same advantage in winning the MDVS Off-The-Record games of Π_{MDVS}. □

Remark 11. 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 and for suggesting Naor-Yung's scheme [26] together with a Simulation Sound NIZK scheme [28] and a Binding Commitment scheme as a starting point to construct the PKEBC scheme. 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. 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

10. 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
11. 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
12. 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
13. 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
14. 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
15. 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
16. 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
17. 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
18. 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
19. 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)
20. 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
21. 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
22. 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
23. 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

24. 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. LNCS, vol. 13092, pp. 189–219. Springer (2021), https://doi.org/10.1007/978-3-030-92078-4_7
25. Maurer, U., Portmann, C., Rito, G.: Multi-designated receiver signed public key encryption. Cryptology ePrint Archive, Report 2022 (2022), https://eprint.iacr.org/2022
26. 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
27. 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
28. 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
29. 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
30. 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
31. 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