On the Relationship Between Weak and Strong Deniable Authenticated Encryption

Kasper Rasmussen
Department of Computer Science
University of Oxford, United Kingdom

Paolo Gasti
Department of Computer Science
New York Institute of Technology

***Abstract*—Consider a scenario in which a whistleblower (Alice) would like to disclose confidential documents to a journalist (Bob). Bob wants to verify that the messages he receives are *really* from Alice; at the same time, Alice does not want to be implicated if Bob is later compelled to (or decides to) disclose her messages, together with his secret key and any other relevant secret information. To fulfill these requirements, Alice and Bob can use a *deniable authenticated* encryption scheme. In this paper we formalize the notions of strong- and weak deniable authentication, and discuss the relationship between these definitions. We show that Bob can still securely authenticate messages from Alice after all his secret information is revealed to the adversary, but *only* when using a weakly (but not strongly) deniable scheme. We refer to this ability as *post-compromise message authentication*. We present two efficient encryption schemes that provide deniable authentication. Both schemes incur overhead similar to that of non-deniable schemes. As such, they are suitable not only when deniability is needed, but also as general encryption tools. We provide details of the encryption, decryption, forgery and key-generation algorithms, and formally prove that our schemes are secure with respect to confidentiality, data authentication, and strong- and weak deniable authentication.**

I. Introduction

In most applications, the recipient of an encrypted message must be able to determine if the message has been tampered with in transit (non-malleability), and if the message was sent by the claimed sender (sender authentication). The former property is commonly achieved using an encryption scheme that provides security against chosen-ciphertext attacks (i.e., a scheme that is CCA-secure [22]). The latter is generally obtained, in the public key setting, using digital signatures. Although combining CCA-secure encryption with signatures normally leads to a secure scheme, the security properties of the resulting construction might be *too strong*, depending on the scenario at hand. Consider, for instance, a whistleblower (Alice) who would like to leak confidential documents to a journalist (Bob). Bob must be able to determine if the messages he received are really from Alice. At the same time, Alice does not want to sign her messages because the adversary (say, a government agency) could compel Bob to disclose the content of his hard drive. If this content includes messages signed by Alice, she would be implicated in the leak.

To meet these requirements, Alice and Bob can use an encryption scheme that provides *deniable* (i.e., non-transferable) authentication. A deniable authenticated encryption scheme allows Bob to verify the authenticity and provenance of messages from Alice. This is important in a whistleblower scenario, because the ability to verify Alice's identity can add to the credibility of the message itself. For instance, receiving a message signed using a key which is certified by the DoD's email CA would give Bob some confidence that the message is really coming from a DoD employee. At the same time, a deniable authenticated encryption scheme gives Bob the ability to create arbitrary fake messages as if they were from Alice. Because Bob is able to generate arbitrary fake messages, Alice's involvement cannot be proven to a third party, i.e., she can *deny* that she sent any particular message. In other words, using a deniable authenticated encryption scheme has no downsides for Alice, because the risk of sending a deniably authenticated message is comparable with the risk associated to sending the same message using a non-authenticated message.

978-1-5386-7493-2/18/$31.00 ©2018 IEEE

Contributions. In this paper we present two efficient encryption schemes that provide deniable authentication. The schemes mainly differ in their security properties. The first offers *strong deniable authentication*: Bob can generate fake messages from Alice without having ever received any message; thus, the existence of a message seemingly from Alice does not imply that she has ever communicated with Bob. The second offers *weak deniable authentication*: after receiving a message from Alice, Bob can construct any number of other arbitrary messages which appear to be from Alice; thus, the existence of a message from Alice with a specific content does not imply that she created it.

We introduce formal definitions of weak- and strong deniable authentication. We then show that our schemes are secure with respect to these definition, while also providing message authentication and confidentiality. We consider both weak- and strong deniable authentication notions meaningful for two reasons: (1) it is possible to construct a weakly deniable authenticated encryption scheme which is *not* strongly deniable (the two definitions are distinct); and (2) it is possible to construct a weakly deniable authenticated scheme which allows Bob to determine whether a particular message has been forged by the adversary using Bob's private keys. However, we prove that it is impossible to construct a strongly deniable scheme with the same property. We refer to this property as *post-compromise* message authentication, because it is relevant to scenarios in which the adversary is able to exfiltrate Bob's

private keys without Bob's knowledge (e.g., when Bob's backups have been compromised). This is particularly relevant in the whistleblower scenario, because the adversary is more likely to target a journalist (Bob) that is known to receive tips from whistleblowers [34], [32], rather than all possible users that might be whistleblowers. The notion of post-compromise message authentication is stronger than the traditional notion of message authentication, because only with the former the adversary has access to the recipients' private keys. Finally we show how to extend our schemes to enable Alice to selectively prove to a third party that she did send a specific message.

Our schemes are built on standard tools and assumptions, and incur overhead similar to that of encryption schemes which do not offer deniability.[1] Moreover, our schemes provide *offline* (non-interactive) deniability, which does not require Alice and Bob to be online at the same time in order to exchange messages.

Organization. The rest of the paper is organized as follows: Section II presents a review of related work. In Section III we introduce our system and adversary model, and in Section IV we define the security properties relevant to deniable authenticated encryption. We describe our deniable authenticated schemes in Section V. The security of our schemes is analyzed in Section VI, and in Section VII we discuss an extension to our deniable authenticated schemes. We conclude in Section VIII.

II. Related Work

There are currently a number of schemes that tackle the problem of denying the content or the existence of previously sent messages. In this section we review techniques that are closely related to our work.

Non-transferable signatures are signatures that can only be verified by the designated party. The first work on non transferable signatures is by Jacobsson et al. [20], and has been followed up by a large body of research on the topic (see, e.g., [25], [1], [6], [21]). Brown and Back [5] showed how some of the properties of [20] can be realized using PGP. The message format in [5] achieves our notion of weak deniable authentication.

Deniably authenticated protocols are multi-message interactive protocols that achieve deniable authentication for one, or both, participants. Di Raimondo and Gennaro coined the terms weak- and strong deniability in this context [33]. These definitions correspond roughly to ours, albeit in the interactive setting. The authors only formally define strong deniability, and leave weak deniability as an informal description. Roughly equivalent definitions have been used in subsequent work by Meng et al. [30], [29], [31]. Tian et al. [35] defined multilevel deniability for one- round authenticated key-exchange protocols. In their work they demonstrate that it is sufficient to prove that the session key is exchanged deniably among parties, and then prove that the protocol transcript is simulatable, to show that a protocol is deniable.

[1] A prototype implementation is available at https://goo.gl/vXoK62.

Other frameworks for deniable authentication protocol have also been proposed, both for IKE [4], [8], [37], and for other protocols [27], [28], [2], [3], [12], [9].

Deniable Authenticated Encryption. In [13], Harn and Ren provide an overview of a deniable authentication mechanism designed for email messages. Weak deniable authentication is achieved through RSA or ElGamal signatures in an unconventional way: instead of signing the hash of a message, as in the traditional hash-and-sign paradigm, the scheme of Harn and Ren applies the signature algorithm directly on the encryption of a random key. The resulting signature is therefore (by design) not existentially unforgeable. Ki et al [23] point out that the deniability of the scheme in [13] cannot be proven if the encryption scheme used to encrypt the random key is treated as a black box. Further, in [23] Ki et al. introduce a weakly deniable scheme, which relies on CDH.

Hwang and Chao [14] defined a deniable authenticated encryption scheme that creates a promise (essentially a signature), that can only be verified by the receiver and does not reveal the identity of the sender to a third party. The scheme uses a variation of Schnorr signatures. Hwang and Chao show that their scheme is unforgeable, and that has properties similar to our definition of strong deniability.

Wang et al. [36] introduced a non-interactive deniable authentication scheme secure in the standard model. In their scheme, the sender uses a one-time signature scheme to authenticate each message, and then encrypts the public key of the signature scheme under a symmetric key obtained using the sender's secret key and the receiver's public key. While the receiver is able to verify whether the signature is correct, the adversary is not able to determine the authenticity of a message because it is not given access to the recipient's secret key in the security model of [36]. Therefore, if the adversary is able to coerce the receiver to disclose his secrets (which is assumed in most of the literature on deniable authenticated encryption), the scheme is not deniable.

Hwang and Sung [15] introduced a deniable authenticated encryption scheme based on promised signcryption: any party can generate a "promised" signcryptext on a particular message; however, only the owner of the signing key can convert the promised signcryptext to a valid signature on the message. The resulting scheme is weakly deniable, but also interactive, which makes it unsuitable for applications such as email.

Later, Hwang et al. [16] improved on [15] by making the scheme non-interactive using a variant of the Schnorr signature scheme to authenticate a random nonce used to encrypt the message. The resulting scheme is strongly deniable. Finally, in [17] Hwang and Chi further extended the scheme in [15] to provide CCA security.

Recently, Li et al. [26] introduced a deniable authenticated encryption scheme secure in the Random Oracle model. The authors formalize the notions of confidentiality and message authentication in a way that is substantially equivalent to ours (see sections IV-A: Message Confidentiality, and IV-B: Message Authentication). However, they do not formally define deniability, and simply show that the recipient can construct

properly distributed messages from the sender to himself without requiring the exchange of any message (intuitively, this satisfies our notion of *strong deniable authentication*—see Section IV-D). The security of [26] reduces to the assumption that the Gap Diffie-Hellman (GDH) problem is hard in the selected group. In contrast, our work relies on the more standard DDH assumption.

Fischlin et al. [10] discuss several deniability notions for encryption, namely: *full deniability*, with which any party can generate fake messages from/to any user; *content deniability*, which allows parties with access to legitimate messages to generate arbitrary fake messages; *context deniability*, which enables any party to build fake evidence of interactions between sender and receiver; *time deniability*, which allows any party with access to a message to produce evidence that the message was sent at a different time; and *source* and *destination deniability*, with which any party can create evidence of interaction between sender and receiver given only the public key of the sender and the receiver, respectively. The authors then map several current protocols to these notions.

Deniable encryption is a type of encryption that has the property that any ciphertext can be decrypted into more than one plaintext. The notion was introduced by Canetti et al. [7]. There are similarities between the notions of deniable authentication and of *receiver-deniable* encryption [7], [24], [11], [18], [19]. In fact, both notions allow the sender to deny computing a particular ciphertext. However, deniable encryption provides deniability to the sender even if the adversary obtains a *trusted* copy of a ciphertext (e.g., by tapping the sender's link), while deniable authentication only provides deniability if the adversary cannot trust the provenance of the ciphertext. Although deniable authenticated encryption is a strictly weaker notion than deniable encryption, we believe that in practice it suffices in most scenarios. Moreover, in contrast with deniable encryption, it allows the design of simpler and more efficient schemes, based on well-established cryptographic tools.

III. System and Adversary Models

In this section we describe the system and adversary model used throughout the paper.

System Model. The model of interaction considered in this paper is illustrated in Figure 1. Alice composes a message using a deniably authenticated scheme and sends it to Bob via an unauthenticated channel. This channel can be any suitable means of communication, including an anonymous messaging board, email, an anonymizing service such as Tor and I2P, or a regular unauthenticated IP network. After Bob receives the message, he attempts to convince a third party that Alice sent it. Bob must be able to authenticate the message to ascertain that it really is from Alice. However he must not be able to convince the adversary that the message has originated from Alice. This must be true even if Bob's private cryptographic material, such as nonces and keys, is compromised (or disclosed willingly). In the rest of this paper we will use the term deniability to cover this notion.

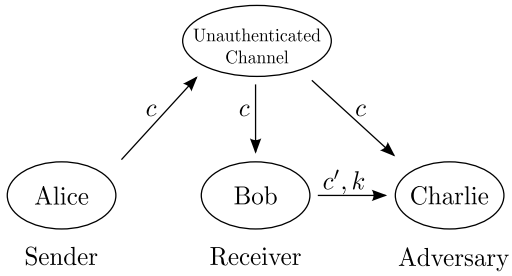

Fig. 1. System model. Alice sends a deniably authenticated message c to Bob through an unauthenticated channel (e.g., an email server, or a public forum). Bob then attempts to prove that he received c' (possibly with $c' \neq c$) from Alice by disclosing it to Charlie, together with all his secrets, represented by k. The adversary can eavesdrop on the channel, although he can observe neither who sent a particular message, nor which messages was received by Bob.

We assume the existence of a public-key infrastructure, trusted by Alice and Bob. We further assume that the existence of a public key corresponding to Alice does not, by itself, incriminate Alice. We argue that this is not a strong assumption, as the public keys used by our schemes are of a similar type as in many current schemes and protocols.

Adversary Model. We do not make any distinction between Bob choosing to reveal Alice's message of his own free will, and Bob being compelled to do so by the adversary. The adversary is able to compel Bob to reveal any message, as well as any secrets, at any time. Further, the adversary is able to obtain messages from the channel (e.g., by downloading a message off a message board), but he is not able to monitor who originally sent the messages or which messages Bob retrieves.

The adversary is not able to prevent Bob from acting, in other words Bob retains his ability to send and receive messages after the adversary is in possession of Bob's secrets. We do not consider Bob trustworthy at any time, and therefore we do not require Bob to perform any task (e.g., deleting messages) to guarantee message deniability. We further assume that the unauthenticated channel between Alice and Bob does not reveal any additional information that can be used to prove that Alice sent a particular message to Bob.

In practice, the message itself may identify Alice depending on the context. For example, in a whistleblowing scenario, it is possible that certain data is accessible exclusively to a small group of people, and thus the fact that Bob is in possession of the data strongly implicates Alice. We make no assumptions on what the data is, other than the fact that Bob's possession of it does not, by itself, provably implicate Alice as the source. Furthermore, it is possible that Bob is sufficiently trusted within a community that his word alone is enough to implicate Alice, without any cryptographic evidence being necessary. This scenario is outside the scope of what deniable authentication, or any other cryptographic tool, can protect against: if Bob's word is trusted without need for any additional proof, he will always be able to implicate anyone in anything. Finally, Alice does not interact with the adversary, and the adversary does not know Alice's identity ahead of time. For this reason he also cannot compel Alice to give up

her secret key. (In our schemes, however, Alice's messages are still deniable even if Alice's secrets are later leaked.) In Section VII we discuss an optional extension allowing Alice to later prove ownership of a message she sent.

IV. Security Definitions

In this section, we formally define the security properties of the proposed schemes. We define confidentiality as a straightforward modification of CCA security, to account for the fact that the sender's secret key and the recipient's identity are used to construct the ciphertext. We also make a similar modification to the definition of message authentication for the same reason. We then formally define weak- and strong deniable authentication, and discuss the relationship between the two notions. We show that no strongly deniable scheme can provide authentication after the receiver's private key has been compromised. To formalize this observation, we first define the notion of post-compromise authentication; we then prove that post-compromise authentication cannot hold for any strongly deniable scheme.

Before we present our security definitions, we specify the list of algorithms that compose a general deniable authenticated scheme. To simplify presentation, we use superscript to denote the entity that is executing the algorithm. For instance, enc^A indicates that algorithm enc is executed by A. When the identity of the party executing the algorithm is either clear from the context, or irrelevant, we omit the superscript. We assume that each algorithm has access to the identity and private key of the entity running it, as well as all public keys. We therefore omit those explicit identity strings and keys from our notation.

- $\mathsf{gen}(1^\kappa, params)$: A probabilistic algorithm that takes as input the security parameter and additional parameters $params$, and returns a public/private keypair.
- $\mathsf{enc}^A(m, B)$: A probabilistic algorithm, executed by A, that takes as input a message m and the identity of the receiver B. It outputs a deniable authenticated ciphertext c.
- $\mathsf{dec}^B(c)$: A deterministic algorithm, executed by B, that takes as input a ciphertext c and outputs a message m and a sender identity A, such that $m = \mathsf{dec}^B(\mathsf{enc}^A(m, B))$, or $\perp$ if the ciphertext is malformed.

A weakly deniable scheme includes a forge algorithm defined as follows:

- $\mathsf{forge}^B_{\mathsf{cnt}}(c^A, m, A)$: A probabilistic algorithm that takes in input $c^A = \mathsf{enc}^A(m', B)$, a message m (possibly $m \neq m'$), and a fake sender identity A. It outputs a new ciphertext c' such that $(m, A) = \mathsf{dec}^B(c')$.

A strongly deniable scheme includes a forge algorithm defined as follows:

- $\mathsf{forge}^B_{\mathsf{ivl}}(m, A)$: A probabilistic algorithm that takes in input a message m and a fake sender identity A. It outputs a new ciphertext c' such that $(m, A) = \mathsf{dec}^B(c')$.

A. Message Confidentiality

As mentioned above, we rely on a slight modification of the standard notion of public-key CCA security (see, e.g., [22]) to accommodate for the fact that the ciphertexts are computed using sender information that is available to the adversary (namely, Alice's private keys and nonces). We provide the adversary with the public keys of both sender (Alice) and receiver (Bob). As in the standard CCA experiment, the adversary is able to query the dec^B oracle. However, because the adversary might not be able to construct ciphertexts from Alice, we also provide oracle access to enc^A. We call the modified notion of CCA security DEN-CCA and it is formalized in Experiment 1.

Experiment 1 ($\text{DEN-CCA}_{\mathcal{ADV},\Pi}(\kappa)$*):* The *CCA indistinguishability of deniable authentication* experiment is composed of the following steps:

1) $\mathcal{ADV}$ is given oracle access to $\mathsf{enc}^A(\cdot, B)$ and $\mathsf{dec}^B(\cdot)$. Eventually $\mathcal{ADV}$ outputs two same-length strings m_0 and m_1.
2) A random bit b is chosen, and $\mathcal{ADV}$ is provided with $c = \mathsf{enc}^A(m_b, B)$.
3) $\mathcal{ADV}$ still has oracle access to enc^A and dec^B. Let $\mathcal{Q}$ be the set of queries made by $\mathcal{ADV}$ to the dec^B oracle in this stage.
4) Eventually, $\mathcal{ADV}$ outputs $b' \in \{0, 1\}$. The experiment outputs 1 if $b = b'$ and $c \notin \mathcal{Q}$, and 0 otherwise.

If the adversary cannot extract any information from ciphertext c, then he cannot guess which of its two messages are encrypted in Step 2 of Experiment 1; that is, he cannot guess b correctly with probability higher than 1/2. This is formalized as DEN-CCA-security in the following definition:

*Definition 1 (*DEN-CCA*-security):* A deniable authenticated encryption scheme $\Pi = (\mathsf{gen}, \mathsf{enc}, \mathsf{dec}, \mathsf{forge} \in \{\mathsf{forge}_{\mathsf{cnt}}, \mathsf{forge}_{\mathsf{ivl}}\})$ has indistinguishable ciphertexts under chosen-ciphertext attack if, for all probabilistic polynomial time adversaries $\mathcal{ADV}$, there exist a negligible function negl such that:

$$\Pr[\text{DEN-CCA}_{\mathcal{ADV},\Pi}(\kappa) = 1] \leq \frac{1}{2} + \mathrm{negl}(\kappa)$$

B. Message Authentication

To formally define message authentication in the context of deniability, we modify the standard notion of existential unforgeability under chosen message attack [22] to account for the non-transferability property of authenticated messages:

Experiment 2 ($\text{DEN-AUTH}_{\mathcal{ADV},\Pi}(\kappa)$*):* The *message forgery of deniable authentication* experiment is composed of the following steps:

1) $\mathcal{ADV}$ is given oracle access to $\mathsf{enc}^A(\cdot, \cdot)$. Let $\mathcal{Q}$ denote the set of pairs (m, B) requested by $\mathcal{ADV}$ as parameters of enc^A in this step.
2) Eventually, $\mathcal{ADV}$ outputs (c, m). The experiment outputs 1 if $\mathsf{dec}^B(c) = (m, A)$, and $(m, B) \notin \mathcal{Q}$, and 0 otherwise.

In other words, $\mathcal{ADV}$ wins the DEN-AUTH experiment if it is able to output a valid ciphertext from Alice to Bob that was never generated by Alice. Moreover, this experiment captures instances in which $\mathcal{ADV}$'s goal is to transform a ciphertext for one recipient into ciphertexts for another different recipient,

because the enc^A oracle answers encryption queries for any recipient. If the adversary is not able to win this experiment (except with negligible probability), we say that the scheme has the message authentication property. This is formalized as Message Authentication in the following definition:

Definition 2 (Message Authentication): A deniable authenticated encryption scheme $\Pi = (\mathsf{gen}, \mathsf{enc}, \mathsf{dec}, \mathsf{forge} \in \{\mathsf{forge}_{\text{cnt}}, \mathsf{forge}_{\text{ivl}}\})$ is existentially unforgeable under chosen-message attack if, for all probabilistic polynomial time adversaries $\mathcal{A}\mathcal{D}\mathcal{V}$, there exist a negligible function negl such that:

$$\Pr[\text{DEN-AUTH}_{\mathcal{A}\mathcal{D}\mathcal{V},\Pi}(\kappa) = 1] \leq \mathrm{negl}(\kappa)$$

C. Weak Deniable Authentication

Informally, if a message has the weak deniable authentication property, it must be impossible for the intended recipient (or anyone else) to prove that the content of the message was created by the sender. This must hold true even if the recipient hands over his private key material and random nonces.

We define weak deniable authentication formally in terms of the deniable content experiment DEN-CNT (Experiment 3), introduced next. In this experiment, the adversary is given access to all Bob's secrets, and must guess whether the ciphertext he received was constructed by Alice, or was forged by Bob.

Experiment 3 ($\text{DEN-CNT}_{\mathcal{A}\mathcal{D}\mathcal{V},\Pi}(\kappa)$*):* The *Deniable Content Experiment* involves Bob and $\mathcal{A}\mathcal{D}\mathcal{V}$, and is composed of the following steps:

1) $\mathcal{A}\mathcal{D}\mathcal{V}$ is provided with Bob's private key $priv_B$.
2) $\mathcal{A}\mathcal{D}\mathcal{V}$ and Bob are given oracle access to $\mathsf{enc}^A(\cdot,\cdot)$. Note that $\mathcal{A}\mathcal{D}\mathcal{V}$ does not need oracle access to dec^B as it already has Bob's private key.
3) $\mathcal{A}\mathcal{D}\mathcal{V}$ eventually outputs two strings m_0 and m_1.
4) Bob selects a random bit $b \in \{0,1\}$, and uses oracle enc^A to obtain ciphertext c_b that encrypts m_b, and sets $c_{\neg b} = \mathsf{forge}^B_{\text{cnt}}(c_b, m_{\neg b}, A)$.
5) Bob sends c_0 and c_1 to $\mathcal{A}\mathcal{D}\mathcal{V}$. Note that if $b = 0$, then c_0 was encrypted by the oracle, while c_1 was forged by Bob (and vice-versa for $b = 1$).
6) Eventually, $\mathcal{A}\mathcal{D}\mathcal{V}$ returns $b' \in \{0,1\}$. The experiment outputs 1 if $b = b'$, i.e., if $\mathcal{A}\mathcal{D}\mathcal{V}$ correctly guesses whether the message was forged by Bob, and 0 otherwise.

If the adversary is not able to distinguish messages that where originally sent by Alice from messages that where forged by Bob, then Bob will have no way to prove to the adversary that a particular message was in fact sent by Alice. Using the above experiment, we formalize weak deniable authentication:

Definition 3 (Weak Deniable Authentication): A deniable authenticated encryption scheme $\Pi = (\mathsf{gen}, \mathsf{enc}, \mathsf{dec}, \mathsf{forge}_{\text{cnt}})$ is weakly deniable if, for all probabilistic polynomial time adversaries $\mathcal{A}\mathcal{D}\mathcal{V}$, there exist a negligible function negl such that:

$$\Pr[\text{DEN-CNT}_{\mathcal{A}\mathcal{D}\mathcal{V},\Pi}(\kappa) = 1] \leq \frac{1}{2} + \mathrm{negl}(\kappa)$$

D. Strong Deniable Authentication

Informally, if a scheme has the strong deniable authentication property, it must be impossible for the intended recipient (or for anyone else) to prove that the sender has sent any message using this scheme. This must hold true even if the recipient reveals his private cryptographic material to the adversary.

We define strong deniable authentication formally in terms of the DEN-IVL experiment (Experiment 4). In this experiment the adversary must guess whether a message received from Bob's was generated by Alice, or was constructed by Bob alone. The adversary has access to all Bob's secrets.

Experiment 4 ($\text{DEN-IVL}_{\mathcal{A}\mathcal{D}\mathcal{V},\Pi}(\kappa)$*):* The *Deniable Involvement* experiment involves Bob and $\mathcal{A}\mathcal{D}\mathcal{V}$, and is composed of the following steps:

1) $\mathcal{A}\mathcal{D}\mathcal{V}$ is provided with Bob's private key $priv_B$.
2) $\mathcal{A}\mathcal{D}\mathcal{V}$ and Bob are given oracle access to $\mathsf{enc}^A(\cdot,\cdot)$. Note that $\mathcal{A}\mathcal{D}\mathcal{V}$ does not need oracle access to dec^B as it already has Bob's private key.
3) $\mathcal{A}\mathcal{D}\mathcal{V}$ selects a message m, and sends it to Bob.
4) Bob selects a random $b \in \{0,1\}$. If $b = 0$, he uses oracle enc^A to obtain ciphertext c which encrypts m. Otherwise, he computes $c = \mathsf{forge}^B_{\text{ivl}}(m, A)$.
5) Bob sends c to $\mathcal{A}\mathcal{D}\mathcal{V}$.
6) Eventually, $\mathcal{A}\mathcal{D}\mathcal{V}$ sends back $b' \in \{0,1\}$. The experiment outputs 1 if $b = b'$, and 0 otherwise.

If the adversary is not able to distinguish a ciphertext created by Alice from one that was forged by Bob, then Bob will have no way to prove to the adversary that Alice has ever been involved in the creation of this ciphertext—let alone that she sent a message with specific content. Using the above experiment, we formalize strong deniable authentication:

Definition 4 (Strong Deniable Authentication): A deniable authenticated encryption scheme $\Pi = (\mathsf{gen}, \mathsf{enc}, \mathsf{dec}, \mathsf{forge}_{\text{ivl}})$ is strongly deniable if, for all probabilistic polynomial time adversaries $\mathcal{A}\mathcal{D}\mathcal{V}$, there exist a negligible function negl such that:

$$\Pr[\text{DEN-IVL}_{\mathcal{A}\mathcal{D}\mathcal{V},\Pi}(\kappa) = 1] \leq \frac{1}{2} + \mathrm{negl}(\kappa)$$

E. Relationship Between Notions of Deniability

Strong deniable authentication implies weak deniable authentication. If Bob is using a strongly deniable scheme in the weakly deniable experiment (Experiment 3), his strategy is as follows: after requesting c_b from the $\mathsf{enc}^A(\cdot,\cdot)$ oracle, he constructs $c_{\neg b} = \mathsf{forge}^B_{\text{ivl}}(m_{\neg b}, A)$ on his own. Because the scheme is strongly deniable, $\mathcal{A}\mathcal{D}\mathcal{V}$ cannot tell which message was created by Bob, and therefore cannot win Experiment 3 with non-negligible advantage over chance.

On the other hand, weak deniability does not imply strong deniability. Evidence of this is from our weakly deniable scheme, presented in Section V-A, which is *not* strongly deniable. In particular, because a ciphertext from Alice to Bob constructed using our weakly deniable scheme contain a valid signature from the sender (see Figure 7), Alice cannot deny having sent *some* message (although she can of course deny the message content).

Even though weak deniability is a strictly weaker notion than strong deniability, there is an important reason for designing weakly deniable schemes: only weak deniable authenticated en-

cryption schemes can provide post-compromise authentication. We define this new notion next.

F. Post-compromise Authentication

A deniable authenticated encryption scheme provides post-compromise authentication if the adversary cannot forge valid messages from Alice to Bob, even when given access to all Bob's secrets. The following experiment captures the ability of Bob to determine whether a ciphertext was forged by $\mathcal{A}\mathcal{D}\mathcal{V}$, or encrypted by Alice, under the assumption that the adversary is unable to drop messages from Alice to Bob. (This assumption is consistent with our system model, presented in Section III.) In contrast with the notion of message authentication presented in Section IV-B, here $\mathcal{A}\mathcal{D}\mathcal{V}$ has (possibly covertly) gained access to Bob's secret keys (hence the term post-compromise).

Experiment 5 ($\text{DEN-COMP-AUTH}_{\mathcal{A}\mathcal{D}\mathcal{V},\Pi}(\kappa)$*):* The *post-compromise authentication* experiment involves Bob and $\mathcal{A}\mathcal{D}\mathcal{V}$, and is composed of the following steps:

1) $\mathcal{A}\mathcal{D}\mathcal{V}$ is provided with Bob's private key $priv_B$.
2) $\mathcal{A}\mathcal{D}\mathcal{V}$ is given oracle access to $\mathsf{enc}^A(\cdot,\cdot)$. For each query to oracle $\mathsf{enc}^A(\cdot,\cdot)$, both $\mathcal{A}\mathcal{D}\mathcal{V}$ and Bob receive the output of the oracle.
3) Eventually, $\mathcal{A}\mathcal{D}\mathcal{V}$ outputs ciphertext c. The experiment outputs 1 if $\mathsf{dec}^B(c) = (m, A)$ for some message m, and 0 otherwise.

In other words, $\mathcal{A}\mathcal{D}\mathcal{V}$ wins the DEN-COMP-AUTH experiment if he is able to output a valid ciphertext from Alice, given that Bob has received all ciphertexts generated by Alice.

Definition 5 (Post-compromise Authentication): A weakly deniable authenticated encryption scheme $\Pi = (\mathsf{gen}, \mathsf{enc}, \mathsf{dec}, \mathsf{forge}_{\mathsf{cnt}})$ has post-compromise message authentication if, for all probabilistic polynomial time adversaries $\mathcal{A}\mathcal{D}\mathcal{V}$, there exist a negligible function negl such that:

$$\Pr[\text{DEN-COMP-AUTH}_{\mathcal{A}\mathcal{D}\mathcal{V},\Pi}(\kappa) = 1] \leq \mathrm{negl}(\kappa)$$

We argue that only weakly deniable authenticated schemes that are not strongly deniable can provide post-compromise authentication. With a strongly deniable authenticated encryption scheme, once the adversary has access to Bob's secret key it can construct ciphertexts using $\mathsf{forge}^B_{\mathsf{ivl}}$. The resulting ciphertexts are indistinguishable from the output of enc^A (as per Definition 4), and their construction requires no interaction with Alice. As such, Bob has no way to differentiate between legitimate messages from Alice and messages generated by the adversary. We formalize this next.

Theorem 1: No strongly deniable encryption scheme (Definition 4) offers post-compromise authentication (Definition 5).

Proof 1: If a scheme provides post-compromise authentication, it is possible to construct the following distinguisher $\mathcal{D}$. $\mathcal{D}$ that takes in input Bob's public and private keys, Alice's public key, and a list $\mathcal{L}$ which includes all ciphertexts computed by oracle enc^A and zero or one ciphertext not computed by the oracle, and outputs 0 if $\mathcal{L}$ includes only ciphertexts computed by the oracle, and 1 otherwise. Further $\mathcal{D}$ must output the correct value with non-negligible advantage over $1/2$.

$\mathcal{D}$ can be used to break strong deniability as follows. In Experiment 4, $\mathcal{A}\mathcal{D}\mathcal{V}$ selects two arbitrary messages m and m'. Then, it makes one query to $\mathsf{enc}^A(m', B)$, obtaining c', and sends m to Bob. Bob returns c, and $\mathcal{A}\mathcal{D}\mathcal{V}$ adds c and c' to $\mathcal{L}$. $\mathcal{A}\mathcal{D}\mathcal{V}$ then invokes $\mathcal{D}$ on pub_B, pri_B, pub_A, and $\mathcal{L}$. If $\mathcal{D}$ outputs 0, then the adversary outputs $b' = 0$ (i.e., $\mathcal{A}\mathcal{D}\mathcal{V}$ concludes that c was computed using oracle enc^A rather than $\mathsf{forge}^B_{\mathsf{ivl}}$). Otherwise, $\mathcal{A}\mathcal{D}\mathcal{V}$ outputs $b' = 1$.

It is easy to see that if $\mathcal{D}$'s output is correct, then $\mathcal{A}\mathcal{D}\mathcal{V}$'s choice of b' is also correct. But because in a strong deniable authenticated encryption scheme $\mathcal{A}\mathcal{D}\mathcal{V}$ has only negligible advantage over $1/2$ to output the correct value of b', then the probability that $\mathcal{D}$ outputs the correct value is also bounded by $1/2 + \mathrm{negl}(\kappa)$. Therefore, $\mathcal{D}$ cannot have non-negligible advantage over $1/2$ for any strongly deniable scheme.

The argument used to prove Theorem 1 does not apply to *strictly* weakly deniable (i.e., *not* strongly deniable) schemes. With a weakly deniable encryption scheme, the adversary computes forgeries using legitimate ciphertexts from Alice. As such, it might be possible to construct a scheme where the relationship between legitimate ciphertexts and the corresponding forgeries is evident to the recipient. This would allow Bob to determine that two ciphertexts that should otherwise be independent are, in fact, related—thus revealing that at least one of the messages is forged.

Simultaneously achieving weak deniability and post-compromised authentication requires Bob to maintain state: because for Bob the output of $\mathsf{forge}_{\mathsf{cnt}}$ is indistinguishable from the output of enc^A (as per Definition 3), a stateless Bob is unable to identify related messages from Alice and from the adversary.

V. Our Schemes

In this section we present two deniable authenticated schemes. The first provides message confidentiality, authentication, and strong deniability. The second scheme provides message confidentiality, authentication, weak deniability, and post-compromise authentication.

Our two schemes are constructed from a CCA-secure public key encryption scheme, treated as a black box. We denote encryption of a message m under Bob's public key pub_B as $\mathsf{E}_{pub_B}(m)$. Analogously, decryption of a ciphertext c using Bob's private key $priv_B$ is denoted as $\mathsf{D}_{priv_B}(c)$.

Additionally, our weakly deniable scheme uses existentially unforgeable cryptographic signatures. Signing a message m using Alice's private (signing) key sig_A is indicated as $\mathsf{sign}_{sig_A}(m)$, while $\mathsf{verify}_{ver_A}(S, m)$ indicates verification of signature S on message m using public verification key ver_A. When signing or verifying a message composed of multiple elements, we assume appropriate encoding of such elements.

A Naïve Approach. Deniable authenticated encryption can be instantiated from standard cryptographic tools as follows. Alice and Bob generate their respective private keys x_A and x_B, and publish the corresponding public keys g^{x_A} and g^{x_B}. Alice encrypts a message m for Bob under key $k_{AB} = (g^{x_B})^{x_A}$ using a CCA secure symmetric encryption scheme (e.g., AES-GCM).

$\mathsf{gen}(1^k, params)$:
1) Extract the description of a group $\mathbb{G}$ from $params$.
2) Select a random element x such that $1 \leq x < |\mathbb{G}|$, and compute $g^x \in \mathbb{G}$
3) Generate keypair for E as $(pub, priv) \leftarrow Gen_{\mathsf{E}}(1^\kappa)$
4) Output $(x, priv)$ and (g^x, pub)

Fig. 2. Strongly deniable scheme (key generation). This algorithm returns two pairs of keys, which are used for message authentication/verification and for encryption/decryption.

$\mathsf{enc}^A(m, B)$:
1) Look-up B's public key g^{x_B}
2) Generate master key for B: $mk_{AB} = (g^{x_B})^{x_A}$
3) Generate token $t_{AB} = \mathrm{MAC}_{mk_{AB}}(m)$
4) Output $c \leftarrow \mathsf{E}_{pub_B}(A, t_{AB}, m)$

Fig. 3. Strongly deniable scheme (encryption).

$\mathsf{dec}^B(c)$:
1) $(A, t_{AB}, m) = \mathsf{D}_{priv_B}(c)$

B now knows the claimed identity of the sender A, the authentication token t_{AB}, and the message m.
2) Look-up A's public key g^{x_A}
3) Generate master key: $mk'_{AB} = (g^{x_A})^{x_B}$
4) Generate token $t'_{AB} = \mathrm{MAC}_{mk'_{AB}}(m)$
5) If $t'_{AB} = t_{AB}$, output (m, A). Otherwise, output $\perp$.

Since only A (or B) could have created t_{AB}, B knows the message is from A.

Fig. 4. Strongly deniable scheme (decryption).

$\mathsf{forge}^B_{\mathsf{ivl}}(m, A)$:
1) Look-up A's public key g^{x_A}
2) Generate master key for B: $mk_{AB} = (g^{x_A})^{x_B}$
3) Generate token $t_{AB} = \mathrm{MAC}_{mk_{AB}}(m)$
4) Output $c \leftarrow \mathsf{E}_{pub_B}(A, t_{AB}, m)$

Fig. 5. Strongly deniable scheme (forge). This algorithm constructs a ciphertext from Alice to Bob encrypting m without knowledge of any of Alice's secrets.

Analogously, Bob decrypts the resulting ciphertext using key k_{AB} computed as $(g^{x_A})^{x_B}$. Intuitively, this scheme is deniably authenticated because Bob can forge ciphertexts from Alice by simply encrypting arbitrary messages under k_{AB}, which he can independently compute. Further, because the underlying encryption scheme is CCA-secure, any modification to c is immediately detected by the recipient. Therefore, for Bob, a valid ciphertext encrypted under k_{AB} is an authenticated message from Alice.

Unfortunately, this scheme is not practical. In order to compute k_{AB}, Bob must know that the sender of a particular ciphertext is Alice. Because no information on the sender can be extracted from the ciphertext prior to decryption, Bob must attempt decryption using all possible public keys from all users. Not only is this very inefficient, but in practice we cannot assume that the list of all public keys that belong to users that might possibly send a message to Bob is available to him.

One could argue that Alice's (plaintext) identity can be distributed along with the ciphertext, and authenticated as additional data using an authenticated encryption with associated data (AEAD) scheme. While this would allow Bob to immediately determine which key to use for decryption, it would also inform all passive adversaries of the (claimed) identity of the sender of a message, hence possibly constituting non-trivial information leakage. In an environment where metadata is routinely collected, or in a scenario where such collection could be employed, this is undesirable.

In what follows, we introduce two schemes that address both issues efficiently.

A. Strongly Deniable Scheme

We present our strongly deniable scheme in figures 2-5. We assume that Alice and Bob have published certificates containing public values (g^{x_A}, pub_A) and (g^{x_B}, pub_B), generated as shown in Figure 2. Further, they are the only parties with knowledge of $(x_A, priv_A)$ and $(x_B, priv_B)$, respectively.

In this scheme, Alice encrypts a message by first generating a master key for Bob as $mk_{AB} = (g^{x_B})^{x_A}$, where g^{x_B} is Bob's public key and x_A is part of Alice's private key. With this master key, Alice can create an authentication token by computing a message authentication code (MAC), on m. The authentication token is encrypted along with the message and Alice's identity, using a CCA-secure encryption scheme. The encryption algorithm is shown in Figure 3.

The decryption algorithm is presented in Figure 4. After receiving a message, Bob decrypts the ciphertext using his public key $priv_B$ and the decryption function $\mathsf{D}_{priv_B}(\cdot)$. Successful decryption of the ciphertext implies that: (1) the encrypted message has not been modified, as $\mathsf{E}_{pub}(\cdot)$ is a CCA-secure encryption scheme and thus provides non-malleability; and (2) the party that encrypted the message had knowledge of a value t_{AB} that could have been generated only by Alice (or by Bob himself). Therefore, Bob can safely consider the message authentic. However, because Bob can forge any message from Alice without Alice's cooperation using $\mathsf{forge}^B_{\mathsf{ivl}}(fake\_msg, A)$ (see Figure 5), a third party with access to the c and all Bob's secrets would not be able to trust that m is authentic.

Ciphertexts constructed with this scheme do not reveal information about the sender, except to the designated receiver, because the sender identity is not accessible prior to decryption.

The scheme in figures 2-5 provides strong deniable encryption (as proven in Section VI). However, any party with access to Bob's secret information (i.e., x_B, or $mk_{AB} = g^{x_A x_B}$) can undetectably send messages to Bob claiming to be from Alice. In the next section we present a scheme that address this issue at the cost of being only weakly deniable.

B. Weakly Deniable Scheme

Our weakly deniable encryption scheme is presented in figures 6-9. To encrypt a message (see Figure 7), Alice first picks a nonce N_A uniformly at random (and therefore independently from the message). The authentication token is constructed using a regular signature scheme as follows. As in the previous scheme, Alice does not authenticate the message directly. Instead, she signs N_A, and the signature is encrypted

$\mathsf{gen}(1^k, params)$:
1) Generate keypair for sign: $(sig, ver) \leftarrow Gen_{\mathsf{sign}}(1^\kappa)$
2) Generate keypair for E: $(pub, priv) \leftarrow Gen_{\mathsf{E}}(1^\kappa)$
3) Output $(sig, priv)$ and (ver, pub)

Fig. 6. Weakly deniable scheme (key generation). This algorithm returns two pairs of keys, which are used for message authentication/verification and for encryption/decryption.

$\mathsf{enc}^A(m, B)$:
1) Pick a nonce $N_A \leftarrow \{0,1\}^k$
2) $t_{AB} \leftarrow \mathsf{sign}_{sig_A}(N_A, B)$
3) Output $c \leftarrow \mathsf{E}_{pub_B}(A, N_A, t_{AB}, m)$

Fig. 7. Weakly deniable scheme (encryption).

together with the data to provide authentication. (Note that the signing key used by Alice to authenticate messages must be used exclusively in this scheme. If the same key is used by Alice to sign arbitrary messages, an adversary who is able to play the signature's existential unforgeability experiment can force Alice to sign a fresh nonce, thus breaking authentication in our scheme.)

Decryption, shown in Figure 8, works as follows. Upon receiving a message, Bob decrypts the ciphertext, and in the process verifies that it is unaltered (thanks to the CCA-security of the underlying encryption scheme). This will reveal the claimed identity of the sender, and allow Bob to retrieve Alice's verification key needed to check signature t_{AB}. Additionally, Bob must verify that N_A was not used in any previous message.

In order for Bob to forge a message from Alice, he must first receive a pair (N_A, t_{AB}), as Bob cannot construct t_{AB} on his own. This makes the scheme only weakly deniable, because Bob's possession of this pair proves to the adversary that Alice did, at some point, send a message. However, unlike our strongly deniable scheme, this scheme offers post-compromise message authentication: even if Bob's secrets are compromised, he can still verify signatures created by Alice and thus authenticate any future messages. No offline external adversary can forge signatures from Alice on a fresh nonce, even if it has access to N_A, t_{AB}, and Bob's private key. In practice, detection of reuse of N_A on a new ciphertext, and therefore knowledge that the adversary has access to Bob's private key, should lead to revocation of Bob's key material. In addition, the two messages containing the same nonce must be both discarded.

Because (N_A, t_{AB}) can be computed by Alice without knowledge of the message that will be encrypted, this scheme allows *offline* signature computation. This technique can be used to implement a tradeoff between time of encryption and space needed to store pre-computed values. In addition, this construction uses only standard tools such as signatures and public key encryption, and its security analysis (Section VI) treats these components as black-boxes. For this reason, this scheme is easy to implement, and allows the use of an already deployed PKI for authentication and for confidentiality (e.g., by using RSA signatures and RSA-based hybrid CCA-secure encryption).

$\mathsf{dec}^B(c)$:
1) $(A, N_A, t_{AB}, m) = \mathsf{D}_{priv_B}(c)$

B now has the claimed identity of the sender A, a random value N_A, an authentication token t_{AB} and the message.

2) Look-up A's public key pub_A
3) Check if N_A was received before from A. If not, add it to B's state. Otherwise, output $\perp$ and terminate.
4) If $\mathsf{verify}_{ver_A}(t_{AB}, (N_A, B))$ returns true, output (m, A). Otherwise, output $\perp$.

Since only A can produce a valid signature t_{AB}, B knows the message is from A (assuming N_A has not been used before).

Fig. 8. Weakly deniable scheme (decryption).

$\mathsf{forge}^B_{\mathrm{cnt}}(c, m', A)$:
1) $(A, N_A, t_{AB}, m) = \mathsf{D}_{priv_B}(c)$
2) Output $c' \leftarrow \mathsf{E}_{pub_B}(A, N_A, t_{AB}, m')$

Fig. 9. Weakly deniable scheme (forge). This algorithm constructs a ciphertext from Alice to Bob encrypting m, without knowledge of any of Alice's secrets.

VI. Security Analysis

In this section we show that our schemes provide both confidentiality (Definition 1), and message authentication (Definition 2). In addition, we show that the scheme defined in figures 2-5 provides strong deniable authentication as defined by Definition 4. Similarly, we show that the scheme represented in figures 6-9 provides weak deniable authentication as per Definition 3.

A. Message Confidentiality

Theorem 2 and 3 state that our strongly- and weakly deniable authenticated schemes enjoy message confidentiality (Definition 1). We present only a proof sketch of this property as it follows directly from the CCA-security of the underlying encryption scheme.

Theorem 2: Assuming that E is a CCA-secure public key encryption scheme, our strongly deniable authenticated encryption scheme (figures 2-5) is DEN-CCA-secure.

Proof 2: (Sketch) CCA-security of our scheme follows from the CCA-security of E. In particular, in the DEN-CCA experiment (Experiment 1) the enc oracle is implemented by first generating fresh values x_A and g^{x_B}, used to compute $g^{x_A x_B}$, and then by answering all queries using E with the public key provided by the CCA challenger. The dec oracle is implemented by using the D oracle, and then returning (m, A). Because E is a CCA-secure encryption scheme, the resulting scheme is also CCA-secure.

Theorem 3: Assuming that E is a CCA-secure public key encryption scheme, our weakly deniable authenticated encryption scheme (figures 6-9) is DEN-CCA-secure.

Proof 3: (Sketch) DEN-CCA-security of our scheme follows from the CCA-security of E. In particular, in the DEN-CCA experiment (Experiment 1) the enc oracle is implemented by first generating a fresh signing keypair for A, using it to

compute t_{AB}, and then answering all queries using E with the public key provided by the CCA challenger. The dec oracle is implemented by using the D oracle, and then returning (m, A). Because E is a CCA-secure encryption scheme, the resulting scheme is DEN-CCA-secure.

B. Message Authentication

In this section we prove that our schemes enjoy message authentication according to Definition 2 (theorems 4 and 5).

Theorem 4: Assuming that E is a CCA-secure public key encryption scheme, that MAC is a secure message authentication code, and DDH is hard in the underlying group, our strongly deniable authenticated encryption scheme (figures 2-5) produces authenticated messages secure under Definition 2.

Proof 4: Assume that there exists an efficient adversary $\mathcal{AD}\mathcal{V}$ that can win Experiment 2 with non-negligible probability. We argue that the same adversary can be used by a simulator SIM to break DDH as follows. SIM receives a tuple $\tau = (g, g^a, g^b, g^c)$ from the challenger, and generates keypairs $(pub_A, priv_A)$ and $(pub_B, priv_B)$ for E. It then sends (pub_A, g^a), and (pub_B, g^b) to $\mathcal{AD}\mathcal{V}$. For each encryption query $\mathsf{enc}^A(m_i, B)$, SIM selects a random string of appropriate length and encrypts it under pub_B, obtaining c_i. SIM stores (c_i, m_i) in a table and uses them to answer decryption queries. Because E is CCA-secure, $\mathcal{AD}\mathcal{V}$ cannot tell that it is the encryption of a random string. Eventually, $\mathcal{AD}\mathcal{V}$ outputs (c, m) such that $\mathsf{dec}^B(c) = (m, A)$ and $\mathsf{enc}^A(m, B)$ was not queried before by $\mathcal{AD}\mathcal{V}$. After decrypting c using $priv_B$, SIM learns t_{AB}. SIM then computes $s = \mathrm{MAC}_{(g^c)}(m)$. SIM learns that τ is a Diffie-Hellman tuple if $s = t_{AB}$, and that τ is not a Diffie-Hellman tuple otherwise. In fact, because MAC is a secure message authentication code, $s = t_{AB}$ iff $\mathrm{MAC}_{(g^c)}(m) = \mathrm{MAC}_{(g^{a \cdot b})}(m)$, which implies that $g^c = g^{a \cdot b}$ (except with negligible probability), i.e., $\mathcal{AD}\mathcal{V}$ can only generate t_{AB} if it can compute $g^{a \cdot b}$ from g^a and g^b. This clearly violates the assumption that DDH is hard in the underlying group, and therefore $\mathcal{AD}\mathcal{V}$ can output a valid pair (c, m) with only negligible probability.

Theorem 5: Assuming that E is a CCA-secure public key encryption scheme, and that sign is an existentially unforgeable signature scheme, our weakly deniable authenticated encryption scheme (figures 6-9) produces authenticated messages secure under Definition 2.

Proof 5: Assume that there exist an efficient adversary $\mathcal{AD}\mathcal{V}$ that can win Experiment 2 with non-negligible probability. We argue that the same adversary can be used by a simulator SIM to break the underlying signature algorithm as follows. SIM receives public key pk for sign from the existential unforgeability challenger, and generates a fresh keypair $pub_B, priv_B$ for E. It then sets $pub_A = pk$ and sends pub_A, pub_B to $\mathcal{AD}\mathcal{V}$. For each encryption query with recipient X, SIM selects a random N_A, asks the challenger to sign pair (N_A, X), obtaining t_{AX}. If $X \neq B$, then SIM generates a fresh public/private key for X, and uses it to compute c. Otherwise, it uses pub_B to compute c as in Figure 7. Eventually, $\mathcal{AD}\mathcal{V}$ outputs (c, m) such that $\mathsf{dec}^B(c) = (m, A)$ and $\mathsf{enc}^A(m, B)$ was not queried before. Because of Theorem 3, $\mathcal{AD}\mathcal{V}$ does not learn any of the values N_A generated by SIM to answer encryption queries. Therefore, after decrypting c using $priv_B$, SIM learns a pair (t^*_{AB}, N^*_A) that was not queried to the challenger with overwhelming probability. This contradicts the existential unforgeability of the underlying signature scheme.

C. Strong Deniable Authentication

In this section we prove that the scheme in figures 2-5 are secure according to the definition of strong deniable authentication (Definition 4).

Theorem 6: Our strongly deniable authenticated encryption scheme (figures 2-5) is strongly deniable per Definition 4.

Proof 6: Assume that there exist an efficient adversary $\mathcal{AD}\mathcal{V}$ that can win Experiment 4 with non-negligible advantage over 1/2. This implies that we can use $\mathcal{AD}\mathcal{V}$ to build a simulator SIM that can distinguish the output of $\mathsf{forge}^B_{\mathsf{ivl}}(m, A)$ and $\mathsf{enc}^A(m, B)$. However this is not possible: $\mathsf{enc}^A(m, B)$ and $\mathsf{forge}^B_{\mathsf{ivl}}(m, A)$ generate the same mk_{AB} (and therefore t_{AB}). Because the resulting input of E in enc and $\mathsf{forge}_{\mathsf{ivl}}$ is identical, the distribution of the output of E in the two functions is also identical. Therefore, our scheme is strongly deniable.

D. Weak Deniable Authentication

We now prove that the scheme in figures 6-9 are secure according to the definition of strong deniable authentication (Definition 3).

Theorem 7: Our weakly deniable authenticated encryption scheme (figures 6-9) is weakly deniable per Definition 3.

Proof 7: Assume that there exist an efficient adversary $\mathcal{AD}\mathcal{V}$ that can win Experiment 4 with non-negligible advantage over 1/2. This implies that we can use $\mathcal{AD}\mathcal{V}$ to build a simulator SIM that can distinguish the output of $\mathsf{forge}^B_{\mathsf{cnt}}(c, m, A)$ and $\mathsf{enc}^A(m, B)$. However this is not possible. In Experiment 3, $\mathcal{AD}\mathcal{V}$ is given $c_0 = \mathsf{E}_{pub_B}(A, N_A, t_{AB}, m_0)$ and $c_1 = \mathsf{E}_{pub_B}(A, N_A, t_{AB}, m_1)$. (One of these values is computed using enc^A, while the other is obtained using $\mathsf{forge}^B_{\mathsf{cnt}}$.) Clearly, the distribution of c_0 and c_1 does not depend on b, and therefore our scheme is strongly deniable.

E. Post-compromise authentication

Theorem 8: Assuming that sign is an existentially unforgeable signature scheme, our weakly deniable authenticated encryption scheme (figures 6-9) has post-compromise message authentication per Definition 5.

Proof 8: The post-compromise authentication of our weakly deniable scheme follows from the existential unforgeability of sign. Assume that there exist an efficient adversary $\mathcal{AD}\mathcal{V}$ that can win Experiment 5 with non-negligible probability. Then $\mathcal{AD}\mathcal{V}$ can be used by a simulator SIM to forge signatures from sign as follows. SIM replies to each query $\mathsf{enc}^A(m_i, B)$ to the enc^A oracle by picking a random $N_{A,i}$ and requesting the existential unforgeability challenger to sign $(N_{A,i}, B)$. Then, $N_{A,i}$ is added to SIM's state. The rest of the ciphertext is computed as shown in Figure 7, Step 3. Ciphertexts computed by SIM are therefore properly distributed.

Let (c, m) be the output of $\mathcal{AD}\mathcal{V}$ at the end of Experiment 5, such that the experiment outputs 1, i.e., $\mathsf{dec}^B(c) = (m, A)$.

SIM computes $(A, N_A, t_{AB}, m) = \mathsf{D}_{priv_B}(c)$, and outputs (N_A, B) and t_{AB} as a valid pair message/signature. Because $\mathsf{D}_{priv_B}(c) \neq \perp$, we know that $N_A \neq N_{A,i}$ for all i. Therefore, the signature of (N_A, B) was never requested to the existential unforgeability challenger. As such, (N_A, B) and t_{AB} represent a valid forgery for sign.

VII. Claiming Ownership of Deniable Authenticated Messages

Our schemes are designed to allow Alice to deny that she sent one or more messages to Bob. However there are circumstances in which Alice might want to be able claim authorship of a message at a later point in time. Such a construction could be useful if, for example, the whistleblowing activities of Alice become generally thought of as heroic because they expose some crime committed by an organization.

Another example is the use of our schemes to implement a simple online auction protocol where all bids, except for the winning one, are deniable. If the bidders encrypt their bids using our schemes, the auction server (Bob) can determine whether a bid is legitimate, but cannot convince any of the bidders that a particular bid is authentic. At the end of the bidding phase, the winner would be asked to claim her bids by publicly proving its authenticity.

Our schemes can be modified to allow this functionality as follows. The message component is augmented with the encryption of a signature on the message itself. The signature is issued using a certified keypair, where the signing key is known only to the bidder. The encryption key used to encrypt the signature is a fresh symmetric key, unknown to any other party. This serves as a commitment by the bidder to the message, and cannot be verified until after the auction, when the bidder publishes the corresponding symmetric key. By publishing this key, the bidder opens the commitment, and proves that she is the author of the message. However if she deletes the encryption key, then no one can prove that she sent that message.

VIII. Conclusion

In this paper we have formalized two notions of deniable authenticated encryption—strong deniability, and weak deniability—and discussed their relationship. We introduced the notion of post-compromise message authentication, which allows a recipient whose private keys have been (possibly secretly) acquired by the adversary to determine whether a deniably authenticated ciphertext is authentic. We proved that this security property can be achieve by a weakly deniable scheme, but not by any strongly deniable construction. To our knowledge, no prior deniable authenticated encryption scheme guarantees post-compromise message authentication.

We then introduced two novel deniable authenticated encryption schemes that are widely applicable as general encryption tools. Further, their deniability features make them well suited to whistleblowing. Our constructions are substantially simpler than previous deniable authenticated encryption schemes. This makes their security analysis simpler, and their implementation less error-prone, compared to the state of the art.

References

[1] Giuseppe Ateniese et al. Sanitizable signatures. In *European Symposium on Research in Computer Security*. Springer, 2005.
[2] Joseph Bonneau et al. Finite-State Security Analysis of OTR V. 2, 2006.
[3] Nikita Borisov et al. Off-the-record communication, or, why not to use PGP. In *WPES*, 2004.
[4] Colin Boyd et al. Deniable authenticated key establishment for internet protocols. *Security Protocols*, 2005.
[5] Ian Brown et al. Non-transferable signatures with PGP.
[6] Christina Brzuska et al. Security of sanitizable signatures revisited. In *PKC 2009*. Springer Berlin Heidelberg, 2009.
[7] Ran Canetti Cynthia et al. Deniable encryption. In *Crypto*, 1997.
[8] Mario Di Raimondo et al. Deniable authentication and key exchange. *CCS '06*, 2006.
[9] Mario Di Raimondo et al. New approaches for deniable authentication. *Journal of Cryptology*, may 2009.
[10] Marc Fischlin et al. Notions of deniable message authentication. In *WPES*. ACM, 2015.
[11] Paolo Gasti, Giuseppe Ateniese, and Marina Blanton. Deniable cloud storage: sharing files via public-key deniability. In *WPES*, 2010.
[12] Ian Goldberg et al. Multi-party off-the-record messaging categories and subject descriptors. In *CCS*, 2009.
[13] Lein Harn and Jian Ren. Design of fully deniable authentication service for e-mail applications. *IEEE Communications Letters*, 2008.
[14] Shin-Jia Hwang et al. An efficient non-interactive deniable authentication protocol with anonymous sender protection. *Journal of Discrete Mathematical Sciences and Cryptography*, 2010.
[15] Shin-Jia Hwang et al. Confidential deniable authentication using promised signcryption. *Journal of Systems and Software*, 2011.
[16] Shin-Jia Hwang et al. *Deniable Authentication Protocols with Confidentiality and Anonymous Fair Protections*. Springer, 2013.
[17] Shin-Jia Hwang et al. Non-interactive fair deniable authentication protocols with indistinguishable confidentiality and anonymity. *Journal of Applied Science and Engineering*, 2013.
[18] M. Ibrahim. A method for obtaining deniable public-key encryption. *International Journal of Network Security*, 2009.
[19] M. Ibrahim. Receiver-deniable public-key encryption. *International Journal of Network Security*, 2009.
[20] Markus Jakobsson et al. Designated verifier proofs and their applications. *Lecture Notes in Computer Science*, 1996.
[21] Robert Johnson et al. Homomorphic signature schemes. In *CT-RSA*, CT-RSA 2002. Springer-Verlag, 2002.
[22] Jonathan Katz et al. *Introduction to Modern Cryptography: Principles and Protocols*. CRC Press, 2007.
[23] JuHee Ki et al. Privacy-enhanced deniable authentication e-mail service. In *DEIS*, 2011.
[24] M. Klonowski et al. Practical deniable encryption. In *SPFSEM'08*, LNCS, 2008.
[25] Hugo Krawczyk et al. Chameleon hashing and signatures. *IACR Cryptology ePrint Archive*, 1998.
[26] Fagen Li et al. Efficient deniably authenticated encryption and its application to e-mail. *IEEE T-IFS*, 2016.
[27] Bo Meng. Formalizing deniability. *ITJ*, 2009.
[28] Bo Meng. A secure non-interactive deniable authentication protocol with strong deniability based on discrete logarithm problem and its application on internet voting protocol. *ITJ*, 2009.
[29] Bo Meng. Automatic verification of deniable authentication protocol in a probabilistic polynomial calculus with cryptoverif. *ITJ*, 2011.
[30] Bo Meng et al. Automatic proofs of deniable authentication protocols with a probabilistic polynomial calculus in computational model. *International Journal of Digital Content Technology and its Applications*, jan 2011.
[31] Bo Meng et al. Computationally sound mechanized proofs for deniable authentication protocols with a probabilistic polynomial calculus in computational model. *ITJ*, 2011.
[32] Nsa hack compromises al-jazeera sources us credibility. https://cl.ly/r46H.
[33] Mario Di Raimondo et al. New approaches for deniable authentication. In *Eurocrypt*, 2006.
[34] NSA targeted journalists critical of government. https://cl.ly/r3iX.
[35] Haibo Tian et al. Deniability and forward secrecy of one-round authenticated key exchange. *The Journal of Supercomputing*, jun 2013.
[36] Bin Wang et al. A non-interactive deniable authentication scheme in the standard model. *IACR Cryptology ePrint Archive*, 2011.
[37] AC Yao et al. Deniable internet key exchange. *ACNS*, 2010.