Strengthening Access Control Encryption*

Christian Badertscher, Christian Matt, and Ueli Maurer
Department of Computer Science, ETH Zurich, 8092 Zurich, Switzerland.
{badi, mattc, maurer}@inf.ethz.ch

Abstract

Access control encryption (ACE) was proposed by Damgård et al. to enable the control of information flow between several parties according to a given policy specifying which parties are, or are not, allowed to communicate. By involving a special party, called the *sanitizer*, policy-compliant communication is enabled while policy-violating communication is prevented, even if sender and receiver are dishonest. To allow outsourcing of the sanitizer, the secrecy of the message contents and the anonymity of the involved communication partners is guaranteed.

This paper shows that in order to be resilient against realistic attacks, the security definition of ACE must be considerably strengthened in several ways. A new, substantially stronger security definition is proposed, and an ACE scheme is constructed which provably satisfies the strong definition under standard assumptions.

Three aspects in which the security of ACE is strengthened are as follows. First, CCA security (rather than only CPA security) is guaranteed, which is important since senders can be dishonest in the considered setting. Second, the revealing of an (unsanitized) ciphertext (e.g., by a faulty sanitizer) cannot be exploited to communicate more in a policy-violating manner than the information contained in the ciphertext. We illustrate that this is not only a definitional subtlety by showing how in known ACE schemes, a single leaked unsanitized ciphertext allows for an arbitrary amount of policy-violating communication. Third, it is enforced that parties specified to receive a message according to the policy cannot be excluded from receiving it, even by a dishonest sender.

1 Introduction

1.1 Access Control Encryption—Model and Security Requirements

The concept of *access control encryption (ACE)* was proposed by Damgård, Haagh, and Orlandi [DHO16] in order to enforce information flow using cryptographic tools rather than a standard access control mechanism (e.g., a reference monitor) within an information system. If the encryption scheme provides certain operations (e.g., ciphertext sanitization) and satisfies an adequate security definition, then the reference monitor can be outsourced, as a component called the *sanitizer*, to an only partially trusted service provider. The goal of ACE is that the sanitizer learns nothing not intrinsically necessary. Security must also be guaranteed against dishonest users, whether senders or receivers of information, and against certain types of sanitizer misbehavior.

*This is the full version of the article published by Springer-Verlag in the proceedings of ASIACRYPT 2017,

The information flow problem addressed by ACE is defined in a context with a set $\mathcal{R}$ of roles corresponding, for example, to different security clearances. Each user in a system can be assigned several roles. For example the users are employees of a company collaborating on a sensitive project, and they need to collaborate and exchange information by sending messages. Since the information is sensitive, which information a party can see must be restricted (hence the term *access control*), even if some parties are dishonest. In the most general form, the specification of which role may send to which other role corresponds to a relation (a subset of $\mathcal{R} \times \mathcal{R}$) or, equivalently, to a predicate $P\colon \mathcal{R} \times \mathcal{R} \to \{0,1\}$, where $s \in \mathcal{R}$ is allowed to communicate to $r \in \mathcal{R}$ if and only if $P(s,r) = 1$. The predicate P is called the *(security) policy*. Typical examples of such policies arise from the Bell-LaPadula [BL73] model where roles are (partially) ordered, and the so-called "no-write-down" rule specifies that it is forbidden for a user to send information to another user with a lower role. Note that for this specific example, the relation is transitive, but ACE also allows to capture non-transitive security policies.

ACE was designed to work in the following setting. Users can communicate anonymously with a sanitizer. If a user wants to send a message, it is encrypted under a key corresponding to the sender's role. Then the ciphertext is sent (anonymously) to the sanitizer who applies a certain sanitization operation and writes the sanitized ciphertext on a publicly readable bulletin board providing anonymous read-access to the users (receivers). Users who are supposed to receive the message according to the policy (and only those users) can decrypt the sanitized ciphertext.

To ensure security in the described setting, the ACE scheme must at least provide the following guarantees:

1. The encryption must assure privacy and anonymity against dishonest receivers as well as the sanitizer, i.e., neither the sanitizer nor dishonest receivers without access allowed by the policy should be able to obtain information about messages or the sender's role.

2. A dishonest sender must be unable to communicate with a (potentially dishonest) receiver, unless this is allowed according to the policy. In other words, the system must not provide covert channels allowing for policy-violating communication.

As usual in a context with dishonest senders, the first goal requires security against chosen-ciphertext attacks (CCA) because dishonest users can send a ciphertext for which they do not know the contained message and by observing the effects the received message has on the environment, potentially obtain information about the message. This corresponds to the availability of a decryption oracle, as in the CCA-security definition.

Note that the second goal is only achievable if users cannot directly write to the repository or communicate by other means bypassing the sanitizer, and if the sanitizer is not actively dishonest because a dishonest sanitizer can directly write any information received from a dishonest sender to the repository. The assumption that a user cannot bypass the sanitizer and communicate to another party outside of the system can for example be justified by assuming that users, even if dishonest, want to avoid being caught communicating illegitimately, or if only a user's system (not the user) is corrupted, and the system can technically only send message to the sanitizer.

Since the sanitizer is not fully trusted in our setting, one should consider the possibility that an unsanitized ciphertext is leaked (intentionally or unintentionally) to a dishonest party. This scenario can be called *(unsanitized) ciphertext-revealing attack*. Obviously, all information contained in this ciphertext gets leaked to that party. While this cannot be avoided, such an attack should not enable dishonest parties to violate the security requirements beyond that.

We point out that previously proposed encryption techniques (before ACE), such as attribute-based encryption [SW05; GPSW06] and functional encryption [BSW11], enable the design of schemes where a sender can encrypt messages such that only designated receivers (who possess the required key) can read the message. This captures the access control aspects of *read* permissions, but it does not allow to capture the control of *write/send* permissions. In other words, such schemes only achieve the first goal listed above, not the second one.

1.2 Contributions of this Paper

While the proposal of the ACE-concept and of efficient ACE-schemes were important first steps toward outsourcing access control, the existing security definition turns out to be insufficient for several realistic attack scenarios. The main contributions of this paper consist of uncovering issues with existing definitions and schemes, fixing these issues by proposing stronger security notions, and constructing a scheme satisfying our stronger notions.

Issues with existing definitions and schemes. As argued above, chosen-ciphertext attacks should be considered since the use case for ACE includes dishonest senders. Existing definitions, however, do not take this into account, i.e., the adversary does not have access to a decryption oracle in the security games.

Furthermore, existing notions do not consider ciphertext-revealing attacks. Technically speaking, the security game that is supposed to prevent dishonest senders from transmitting information to dishonest receivers (called no-write game), gives the adversary only access to an encryption oracle that sanitizes ciphertexts before returning them. This means that the adversary has no access to unsanitized ciphertexts. This is not only a definitional subtlety, but can completely break down any security guarantees. We demonstrate that existing ACE schemes allow the following attack: Assume there are three users A, M, and E in the system, where A is honest and by the policy allowed to send information to E, and M and E are dishonest and not allowed to communicate. If A sends an (innocent) message to E and the corresponding unsanitized ciphertext is leaked to M, malleability of the ciphertext can be exploited by M to subsequently communicate an arbitrary number of arbitrary messages chosen by M to E. Note that while this attack crucially exploits malleability of ciphertexts, it is not excluded by CCA security for two reasons: first, CCA security does not prevent an adversary from producing valid ciphertexts for *unrelated* messages, and second, the integrity should still hold if the adversary has the decryption key (but not the encryption key).

Finally, existing security definitions focus on preventing dishonest parties from communicating if disallowed by the policy, but they do not enforce information flow. For example, if user A only has a role such that according to the policy, users B and C can read what A sends, existing schemes do not prevent A from sending a message that can be read by B but not by C, or sending a message such that B and C receive different messages. This is not as problematic as the two issues above, and one can argue that A could anyway achieve something similar by additionally encrypting the message with another encryption scheme. Nevertheless, for some use cases, actually precisely enforcing the policy can be required (consider, e.g., a logging system), and one might intuitively expect that ACE schemes achieve this.

New security definitions. We propose new, stronger security definitions for ACE that exclude all issues mentioned above. First, we give the adversary access to a decryption oracle. More

precisely, the oracle first sanitizes the given ciphertext and then decrypts it, since this is what happens in the application if a dishonest party sends a ciphertext to the sanitizer. Second, we incorporate ciphertext-revealing attacks by giving the adversary access to an encryption oracle that returns unsanitized ciphertexts for arbitrary roles. Finally, we introduce a new security game in which an adversary can obtain encryption keys and decryption keys from an oracle and has to output a ciphertext such that one of the following events occur: either the set of roles that can successfully decrypt the ciphertext (to an arbitrary message) is inconsistent with the policy for all sender roles for which the adversary has an encryption key (in this case, we say the adversary is not *role-respecting*); or the ciphertext can be successfully decrypted with two keys such that two different messages are obtained (in this case, we say the *uniform-decryption* property is violated).

Construction of an ACE scheme for our stronger notions. Our construction proceeds in three steps and follows the general structure of the generic construction by Fuchsbauer et al. [FGKO17]. Since we require much stronger security notions in all three steps, our constructions and proofs are consequently more involved than existing ones. First, we construct a scheme for a primitive we call *enhanced sanitizable public-key encryption (sPKE)*. Second, we use an sPKE scheme to construct an ACE scheme satisfying our strong security notion for the equality policy, i.e., for the policy that allows s to send to r if and only if $r = s$. Third, we show how to lift an ACE scheme for the equality policy to an ACE scheme for the disjunction of equalities policy. This policy encodes roles as vectors $\mathbf{x} = (x_1, \ldots, x_\ell)$ and allows role $\mathbf{x}$ to send to role $\mathbf{y}$ if and only if $x_1 = y_1 \vee \ldots \vee x_\ell = y_\ell$. As shown by Fuchsbauer et al. [FGKO17], useful policies including the inequality predicate corresponding to the Bell-LaPadula model can efficiently be implemented using this policy by encoding the roles appropriately.

Enhanced sanitizable PKE. An sPKE scheme resembles publicy-key encryption with an additional setup algorithm that outputs sanitizer parameters and a master secret key. The master secret key is needed to generate a public/private key pair and the sanitizer parameters can be used to sanitize a ciphertext. A sanitized ciphertext cannot be linked to the original ciphertext without the decryption key. We require the scheme to be CCA secure (with respect to a sanitize-then-decrypt oracle) and anonymous. Sanitization resembles rerandomization [Gro04; PR07], also called universal re-encryption [GJJS04], but we allow sanitized ciphertexts to be syntactically different from unsanitized ciphertexts. This allows us to achieve full CCA security, which is needed for our ACE construction and unachievable for rerandomizable encryption.

Our scheme is based on ElGamal encryption [Elg85], which can easily be rerandomized and is anonymous. We obtain CCA security using the technique of Naor and Yung [NY90], i.e., encrypting the message under two independent keys and proving in zero-knowledge that the ciphertexts are encryptions of the same message, which was shown by Sahai to achieve full CCA security if the zero-knowledge proof is simulation-sound [Sah99]. A technical issue is that if the verification of the NIZK proof was done by the decrypt algorithm, the sanitization would also need to sanitize the proof. Instead, we let the sanitizer perform the verification. Since we want to preserve anonymity, this needs to be done without knowing under which public keys the message was encrypted. Therefore, the public keys are part of the witness in the NIZK proof. Now the adversary could encrypt the same message under two different public keys that were not produced together by the key-generation, which would break the reduction. To prevent this, the pair of public keys output by the key-generation is signed using a signature key that is

contained in the master secret key and the corresponding verification key is contained in the sanitizer parameters.

ACE for equality. The basic idea of our ACE scheme for the equality policy is to use for each role, encryption and decryption keys of an sPKE scheme as the encryption and decryption keys of the ACE scheme, respectively. Since we need to prevent dishonest senders without an encryption key for some role from producing valid ciphertexts for that role even after seeing encryptions of other messages for this role and obtaining encryption keys for other roles, we add a signature key to the encryption key, sign this pair using a separate signing key, where the corresponding verification key is part of the sanitizer parameters, and let senders sign their ciphertexts. To preserve anonymity, this signature cannot be part of the ciphertext. Instead, senders prove in zero-knowledge that they know such a signature and that the encryption was performed properly.

ACE for disjunction of equalities. The first step of our lifting is identical to the lifting described by Fuchsbauer et al. [FGKO17]: for each component of the role-vector, the encryption and decryption keys contain corresponding keys of an ACE scheme for the equality policy. To encrypt a message, this message is encrypted under each of the key-components. In a second step, we enforce role-respecting security with the same trick we used in our ACE scheme for equality; that is, we sign encryption key-vectors together with a signing key for that role, and senders prove in zero-knowledge that they have used a valid key combination to encrypt and that they know a signature of the ciphertext vector.

1.3 Related Work

The concept of access control encryption has been introduced by Damgård et al. [DHO16]. They provided the original security definitions and first schemes. Subsequent work by Fuchsbauer et al. [FGKO17], by Tan et al. [TZMT17], and by Kim and Wu [KW17] focused on new schemes that are more efficient, based on different assumptions, or support more fine grained access control policies. In contrast to our work, they did not attempt to strengthen the security guarantees provided by ACE.

2 Preliminaries

2.1 Notation

We write $x \leftarrow y$ for assigning the value y to the variable x. For a finite set X, $x \twoheadleftarrow X$ denotes assigning to x a uniformly random value in X. For $n \in \mathbb{N}$, we use the convention

$$[n] \coloneqq \{1, \ldots, n\}.$$

By $\mathbb{Z}_n$ we denote the ring of integers modulo n, and by $\mathbb{Z}_n^*$ its multiplicative group of units. The probability of an event A in an experiment E is denoted by $\Pr^E[A]$, e.g., $\Pr^{x \twoheadleftarrow \{0,1\}}[x = 0] = \frac{1}{2}$. If the experiment is clear from the context, we omit the superscript. The conditional probability of A given B is denoted by $\Pr[A \mid B]$ and the complement of A is denoted by $\neg A$. For a probabilistic algorithm $\mathcal{A}$ and $r \in \{0,1\}^*$, we denote by $\mathcal{A}(x; r)$ the execution of $\mathcal{A}$ on input x with randomness r. For algorithms $\mathcal{A}$ and $\mathcal{O}$, $\mathcal{A}^{\mathcal{O}(\cdot)}(x)$ denotes the execution of $\mathcal{A}$ on input x, where $\mathcal{A}$ has oracle access to $\mathcal{O}$.

2.2 Security Definitions, Advantages, Efficiency, and Negligibility

We define the security of a scheme via a random experiment (or game) involving an adversary algorithm $\mathcal{A}$. For a given scheme $\mathcal{E}$ and adversary $\mathcal{A}$, we define the advantage of $\mathcal{A}$, which is a function of the security parameter κ. To simplify the notation, we omit the security parameter when writing the advantage, e.g., we write $\mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathcal{E},\mathcal{A}}$ instead of $\mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathcal{E},\mathcal{A}}(\kappa)$ for the advantage of $\mathcal{A}$ in the existential unforgeability game for the signature scheme $\mathcal{E}$. Such a scheme is considered *secure* if $\mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathcal{E},\mathcal{A}}$ is *negligible* for all *efficient* $\mathcal{A}$. An algorithm $\mathcal{A}$ is *efficient* if it runs in *probabilistic polynomial time (PPT)*, i.e., $\mathcal{A}$ has access to random bits and there is a polynomial p such that $\mathcal{A}(x)$ terminates after at most $p(|x|)$ steps (on some computational model, e.g., Turing machines) for all inputs x, where $|x|$ denotes the bit-length of x. A function f is *negligible* if for every polynomial p, there exists $n_0 \in \mathbb{N}$ such that $f(n) < 1/p(n)$ for all $n \geq n_0$. While these asymptotic definitions yield concise statements, we will in all proofs derive precise bounds on the advantages, following a concrete security approach.

2.3 Access Control Encryption

We recall the definition of access control encryption by Damgård et al. [DHO16]. For definitions of other cryptographic primitives used in this paper, see Appendix A. Following Fuchsbauer et al. [FGKO17], we do not have sanitizer keys and require Gen to be deterministic. The set of roles is assumed to be $\mathcal{R} = [n]$.

Definition 2.1. An *access control encryption (ACE) scheme* $\mathcal{E}$ consists of the following five PPT algorithms:

Setup: The algorithm Setup on input a security parameter 1^κ and a *policy* $P\colon [n] \times [n] \to \{0,1\}$, outputs a *master secret key* msk and *sanitizer parameters* sp. We implicitly assume that all keys include the finite *message space* $\mathcal{M}$ and the *ciphertext spaces* $\mathcal{C}, \mathcal{C}'$.

Key generation: The algorithm Gen is deterministic and on input a master secret key msk, a role $i \in [n]$, and the type $\mathtt{sen}$, outputs an *encryption key* ek_i; on input msk, $j \in [n]$, and the type $\mathtt{rec}$, outputs a *decryption key* dk_j.

Encryption: The algorithm Enc on input an encryption key ek_i and a message $m \in \mathcal{M}$, outputs a ciphertext $c \in \mathcal{C}$.

Sanitization: The algorithm San on input sanitizer parameters sp and a ciphertext $c \in \mathcal{C}$, outputs a *sanitized ciphertext* $c' \in \mathcal{C}' \cup \{\bot\}$.

Decryption: The algorithm Dec on input a decryption key dk_j and a sanitized ciphertext $c' \in \mathcal{C}'$, outputs a message $m \in \mathcal{M} \cup \{\bot\}$; on input dk_j and $\bot$, it outputs $\bot$.

For a probabilistic algorithm $\mathcal{A}$, consider the experiment $\mathsf{Exp}^{\mathsf{ACE\text{-}CORR}}_{\mathcal{E},\mathcal{A}}$ that given a security parameter 1^κ and a policy P, executes $(sp, msk) \leftarrow \mathsf{Setup}(1^\kappa, P)$, $(m, i, j) \leftarrow \mathcal{A}^{\mathsf{Gen}(msk,\cdot,\cdot)}(sp)$, $ek_i \leftarrow \mathsf{Gen}(msk, i, \mathtt{sen})$, and $dk_j \leftarrow \mathsf{Gen}(msk, j, \mathtt{rec})$. We define the *correctness advantage* of $\mathcal{A}$ (for security parameter κ and policy P) as

$$\mathsf{Adv}^{\mathsf{ACE\text{-}CORR}}_{\mathcal{E},\mathcal{A}} := \Pr\big[P(i,j) = 1 \ \wedge \ \mathsf{Dec}\big(dk_j, \mathsf{San}(sp, \mathsf{Enc}(ek_i, m))\big) \neq m\big],$$

where the probability is over the randomness in $\mathsf{Exp}^{\mathsf{ACE\text{-}CORR}}_{\mathcal{E},\mathcal{A}}$ and the random coins of Enc, San, and Dec. The scheme $\mathcal{E}$ is called *correct* if $\mathsf{Adv}^{\mathsf{ACE\text{-}CORR}}_{\mathcal{E},\mathcal{A}}$ is negligible for all efficient $\mathcal{A}$, and *perfectly correct* if $\mathsf{Adv}^{\mathsf{ACE\text{-}CORR}}_{\mathcal{E},\mathcal{A}} = 0$ for all $\mathcal{A}$.

Remark. Correctness of an encryption scheme is typically not defined via a game with an adversary, but by requiring that decryption of an encryption of m yields m with probability 1. This perfect correctness requirement is difficult to achieve for ACE schemes and not necessary for applications because it is sufficient if a decryption error only occurs with negligible probability in any execution of the scheme. Damgård et al. [DHO16] define correctness by requiring that for all m, i, and j with $P(i,j) = 1$, the probability that a decryption fails is negligible, where the probability is over setup, key generation, encrypt, sanitize, and decrypt. While this definition is simpler than ours, it does not guarantee that decryption errors only occur with negligible probability in any execution of the scheme. For example, a scheme could on setup choose a random message m and embed it into all keys such that decryption always fails for encryptions of this particular message. This does not violate the definition by Damgård et al. since for any fixed message, the probability that this message is sampled during setup is negligible (if the message space is large). Nevertheless, an adversary can always provoke a decryption error by sending that particular message m, which is not desirable. The above example might at first sight seem somewhat artificial, and typically, schemes do not have such a structure. However, capturing correctness via an experiment is important when thinking of composition, since we expect that the correctness guarantee still holds when the ACE scheme is run as part of a larger system. In order to meet this expectation, and to exclude the above issue, we formalize correctness via an experiment.

Additionally, Fuchsbauer et al. have defined detectability, which guarantees that decrypting with a wrong key yields $\perp$ with high probability [FGKO17]. This allows receivers to detect whether a message was sent to them. As for correctness, we define it via an experiment. The notion is related to robustness for public-key encryption [ABN10]. We additionally define strong detectability, in which the randomness for the encryption is adversarially chosen.

Definition 2.2. Let $\mathcal{E} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{San}, \mathsf{Dec})$ be an ACE scheme and let $\mathcal{A}$ be a probabilistic algorithm. Consider the experiment $\mathsf{Exp}^{\mathsf{ACE\text{-}DTCT}}_{\mathcal{E},\mathcal{A}}$ that given a security parameter 1^κ and a policy P, executes $(sp) \leftarrow \mathsf{Setup}(1^\kappa, P)$, $(m,i,j) \leftarrow \mathcal{A}^{\mathsf{Gen}(msk,\cdot,\cdot)}(sp, msk)$, $ek_i \leftarrow \mathsf{Gen}(msk, i, \mathtt{sen})$, and $dk_j \leftarrow \mathsf{Gen}(msk, j, \mathtt{rec})$. We define the *detectability advantage* of $\mathcal{A}$ as

$$\mathsf{Adv}^{\mathsf{ACE\text{-}DTCT}}_{\mathcal{E},\mathcal{A}} := \Pr\big[P(i,j) = 0 \ \wedge \ \mathsf{Dec}\big(dk_j, \mathsf{San}(sp, \mathsf{Enc}(ek_i, m))\big) \neq \perp\big],$$

where the probability is over the randomness in $\mathsf{Exp}^{\mathsf{ACE\text{-}DTCT}}_{\mathcal{E},\mathcal{A}}$ and the random coins of Enc, San, and Dec. The scheme $\mathcal{E}$ is called *detectable* if $\mathsf{Adv}^{\mathsf{ACE\text{-}DTCT}}_{\mathcal{E},\mathcal{A}}$ is negligible for all efficient $\mathcal{A}$. The experiment $\mathsf{Exp}^{\mathsf{ACE\text{-}sDTCT}}_{\mathcal{E},\mathcal{A}}$ is identical to $\mathsf{Exp}^{\mathsf{ACE\text{-}DTCT}}_{\mathcal{E},\mathcal{A}}$ except that $\mathcal{A}$ returns (m,r,i,j). The *strong detectability advantage* of $\mathcal{A}$ is defined as

$$\mathsf{Adv}^{\mathsf{ACE\text{-}sDTCT}}_{\mathcal{E},\mathcal{A}} := \Pr\big[P(i,j) = 0 \ \wedge \ \mathsf{Dec}\big(dk_j, \mathsf{San}(sp, \mathsf{Enc}(ek_i, m; r))\big) \neq \perp\big],$$

where the probability is over the randomness in $\mathsf{Exp}^{\mathsf{ACE\text{-}sDTCT}}_{\mathcal{E},\mathcal{A}}$ and the random coins of San and Dec. The scheme $\mathcal{E}$ is called *strongly detectable* if $\mathsf{Adv}^{\mathsf{ACE\text{-}sDTCT}}_{\mathcal{E},\mathcal{A}}$ is negligible for all efficient $\mathcal{A}$.

2.4 Existing Security Definitions

Existing notions for ACE specify two core properties: the so-called *no-read rule* and the *no-write rule*. The no-read rule formalizes privacy and anonymity: roughly, an honestly generated ciphertext should not leak anything about the message, except possibly its length, or about the

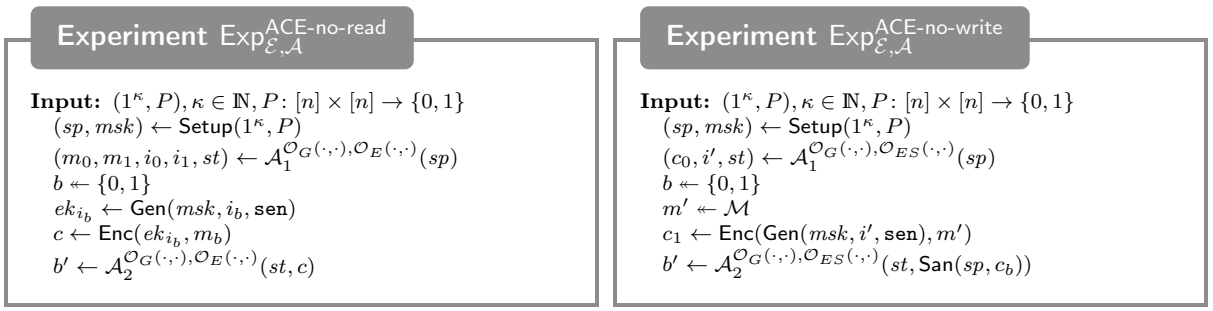

Figure 1: The no-read and no-write experiments for an ACE scheme $\mathcal{E}$ and an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$. The oracles are defined as $\mathcal{O}_G(\cdot,\cdot) := \mathsf{Gen}(msk, \cdot, \cdot)$, $\mathcal{O}_E(\cdot,\cdot) := \mathsf{Enc}(\mathsf{Gen}(msk, \cdot, \mathtt{sen}), \cdot)$, and $\mathcal{O}_{ES}(\cdot,\cdot) := \mathsf{San}(sp, \mathsf{Enc}(\mathsf{Gen}(msk, \cdot, \mathtt{sen}), \cdot))$.

role of the sender. The security game allows an adversary to interact with a key-generation oracle (to obtain encryption and decryption keys for selected roles), and an encryption oracle to obtain encryptions of chosen messages for roles for which the adversary does not possess the encryption key. This attack model reflects that an adversary cannot obtain useful information by observing the ciphertexts that are sent to the sanitizer. To exclude trivial attacks, it is not considered a privacy breach if the adversary knows a decryption key that allows to decrypt the challenge ciphertext according to the policy. Similarly, it is not considered an anonymity breach if the encrypted messages are different. We next state the definition of the no-read rule.[1]

Definition 2.3. Let $\mathcal{E} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{San}, \mathsf{Dec})$ be an ACE scheme and let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a pair of probabilistic algorithms. Consider the experiment $\mathsf{Exp}^{\mathsf{ACE\text{-}no\text{-}read}}_{\mathcal{E},\mathcal{A}}$ in Figure 1 and let J be the set of all j such that $\mathcal{A}_1$ or $\mathcal{A}_2$ issued the query $(j, \mathtt{rec})$ to the oracle $\mathcal{O}_G$. The *payload-privacy advantage* and the *sender-anonymity advantage* of $\mathcal{A}$ are defined as

$$\mathsf{Adv}^{\mathsf{ACE\text{-}no\text{-}read,priv}}_{\mathcal{E},\mathcal{A}} := 2 \cdot \Pr\big[b' = b \ \wedge\ |m_0| = |m_1| \ \wedge\ \forall j \in J\ P(i_0, j) = P(i_1, j) = 0\big] - 1,$$
$$\mathsf{Adv}^{\mathsf{ACE\text{-}no\text{-}read,anon}}_{\mathcal{E},\mathcal{A}} := 2 \cdot \Pr\big[b' = b \ \wedge\ m_0 = m_1 \ \wedge\ \forall j \in J\ P(i_0, j) = P(i_1, j)\big] - 1,$$

respectively, where the probabilities are over the randomness of all algorithms in $\mathsf{Exp}^{\mathsf{ACE\text{-}no\text{-}read}}_{\mathcal{E},\mathcal{A}}$. The scheme $\mathcal{E}$ satisfies the *payload-privacy no-read rule* and the *sender-anonymity no-read rule* if $\mathsf{Adv}^{\mathsf{ACE\text{-}no\text{-}read,priv}}_{\mathcal{E},\mathcal{A}}$ and $\mathsf{Adv}^{\mathsf{ACE\text{-}no\text{-}read,anon}}_{\mathcal{E},\mathcal{A}}$ are negligible for all efficient $\mathcal{A}$, respectively. If it satisfies both, it is said to satisfy the *no-read rule*.

The no-write rule of ACE is the core property to capture access control. In a nutshell, if the adversary only possesses encryption keys for roles i and decryption keys for roles j with $P(i, j) = 0$, then he should not be able to create a ciphertext from which, after being sanitized, he can retrieve any information. Technically, in the corresponding security game, the adversary is given a key-generation oracle as above, and in addition an oracle to obtain *sanitized* ciphertexts for selected messages and roles. This attack model corresponds to a setting where an adversary only sees the outputs of a sanitizer, but not its inputs, and in particular no unsanitized ciphertexts generated for roles for which he does not possess the encryption key. The adversary wins if he manages to distinguish the sanitized version of a ciphertext of his choice from a sanitized

[1] For anonymity, we adopt here the definition of [DHO16], which is stronger than the one used by Fuchsbauer et al. [FGKO17] since there, anonymity is not guaranteed against parties who can decrypt.

version of a freshly generated encryption of a random message, and if he does not obtain the encryption key for any role i and the decryption key of any role j for which $P(i,j) = 1$, as this would trivially allow him to distinguish.

Definition 2.4. Let $\mathcal{E} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{San}, \mathsf{Dec})$ be an ACE scheme and let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a pair of probabilistic algorithms. Consider the experiment $\mathsf{Exp}^{\mathsf{ACE\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}}$ in Figure 1, let I_1 be the set of all i such that $\mathcal{A}_1$ issued the query $(i, \mathtt{sen})$ to $\mathcal{O}_G$, and let J be the set of all j such that $\mathcal{A}_1$ or $\mathcal{A}_2$ issued the query $(j, \mathtt{rec})$ to $\mathcal{O}_G$. We define the *no-write advantage* of $\mathcal{A}$ as

$$\mathsf{Adv}^{\mathsf{ACE\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}} := 2 \cdot \Pr\left[b' = b \ \wedge \ i' \in I_1 \ \wedge \ \forall i \in I_1 \ \forall j \in J \ P(i,j) = 0 \ \wedge \ \mathsf{San}(sp, c_0) \neq \bot\right] - 1,$$

where the probability is over the randomness of all algorithms in $\mathsf{Exp}^{\mathsf{ACE\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}}$. The scheme $\mathcal{E}$ satisfies the *no-write rule* if $\mathsf{Adv}^{\mathsf{ACE\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}}$ is negligible for all efficient $\mathcal{A}$.

Remark. Our definition follows the one by Fuchsbauer et al. [FGKO17] by requiring $\mathsf{San}(sp, c_0) \neq \bot$ in the winning condition for the no-write rule, which was not required in the original definition by Damgård et al. [DHO16]. Schemes can be made secure with respect to the original definition by letting the algorithm San create a fresh ciphertext for a random message when given an invalid ciphertext.

The condition $i' \in I_1$ together with $\forall i \in I_1 \ \forall j \in J \ P(i,j) = 0$ ensures that $\mathcal{A}$ does have a key to decrypt c_1, which would trivially allow to distinguish. Requiring that $\mathcal{A}$ obtains a key for i' however excludes adversaries that obtain no key at all. The original definitions [DHO16] therefore include a special role 0 with $P(0,j) = 0$ for all j. One can then assume without loss of generality that anyone obtains a key for this role. Since assuming the existence of such a role appears to be a technicality that is only needed for the no-write rule, we do not make this assumption and present new security definitions in Section 4.2 that do not rely on such a role.

3 Ciphertext-Revealing Attacks Against Existing Schemes

3.1 Generic Description of Attack

We describe a fundamental practical issue of schemes which meet the above no-read and no-write definitions and show why the guarantees expected from an ACE scheme need to be strengthened. We show that schemes fulfilling the definition can suffer from what we call a malleability attack, which effectively bypasses the given policy and allows communication that is forbidden by the policy. The attack does not abuse any peculiarities of existing models and in fact only requires that the semi-honest sanitizer shares its inputs and outputs with colluding parties, which is arguably possible when the sanitizer is outsourced. In particular, security against such a sanitizer is desirable from a practical point of view.

We first give a high-level explanation of the attack, formalize it as a second step, and show that several existing schemes are vulnerable. Assume there are three parties, Alice, Bob, and Charlie, each having a different role assigned. We denote by A, B, and C the associated roles. In our example, Alice and Charlie are always honest. Alice is allowed to communicate with Bob and Charlie. Bob is dishonest and forbidden to send messages to Charlie (and to Alice). The attack now proceeds as follows: When Alice sends her first message, Bob requests the corresponding ciphertext and the sanitized ciphertext from the semi-honest sanitizer. He then decrypts the sanitized ciphertext and thus receives the message Alice has sent. With the knowledge of this message, as we show below, he is able to create a valid ciphertext for a chosen message m',

which will be correctly sanitized and later decrypted by Charlie, hence allowing unrestricted communication from Bob to Charlie. Details follow.

Consider the policy defined by

$$P(i,j) \coloneqq \begin{cases} 1, & i = \mathsf{A}, \\ 0, & \text{otherwise.} \end{cases}$$

For the sake of presentation, we assume that the ACE scheme $\mathcal{E}$ under consideration enjoys perfect correctness. Also, we assume that the setup-phase has completed and the three parties thus possess the encryption and decryption keys, ek_i and dk_i, respectively. Now, imagine that the ACE scheme admits an efficient function $\mathsf{maul}_{\mathcal{E}}$ with the following property (later we show how to implement such a function for some existing schemes): For all messages m and m', any role i, and sanitizer parameters sp in the range of Setup, and for any fixed randomness r,

$$\mathsf{maul}_{\mathcal{E}}\big(\mathsf{Enc}(ek_i, m; r), sp, m, m'\big) = \mathsf{Enc}(ek_i, m'; r). \tag{1}$$

If such a malleability function exists, the communication policy can be bypassed as follows:

1. Alice encrypts a message $c \leftarrow \mathsf{Enc}(ek_{\mathsf{A}}, m)$ and the sanitizer computes $c' \leftarrow \mathsf{San}(sp, c)$ and gives c and c' to Bob.

2. Bob computes $m \leftarrow \mathsf{Dec}(dk_{\mathsf{B}}, c')$ and creates a new ciphertext $\hat{c} \leftarrow \mathsf{maul}_{\mathcal{E}}(c, sp, m, m')$ and sends it to the sanitizer.

3. The ciphertext is sanitized $\hat{c}' \leftarrow \mathsf{San}(sp, \hat{c})$ and subsequently sent to Charlie. By the (perfect) correctness of the assumed ACE scheme and by our assumption on $\mathsf{maul}_{\mathcal{E}}$, $\hat{c}'$ is a valid ciphertext (under the encryption key of Alice) and Charlie is able to decrypt $m' \leftarrow \mathsf{Dec}(dk_{\mathsf{C}}, \hat{c}')$, effectively receiving Bob's message m'.

In the following sections, we show that several existing ACE schemes $\mathcal{E}$ admit an efficient function $\mathsf{maul}_{\mathcal{E}}$. More specifically, we consider the "linear" scheme by Damgård et al. [DHO16] based on ElGamal and the ElGamal-based scheme by Fuchsbauer et al. [FGKO17].

3.2 DHO Scheme Based on ElGamal

We briefly recall the ElGamal based ACE scheme for a single identity. The sanitizer parameters of the scheme contain the description of a finite cyclic group $G = \langle g \rangle$ and its group order q, and additionally an element $h = g^x$ for a uniform random $x \in \mathbb{Z}_q$. The encryption key for A is a random value $ek \in \mathbb{Z}_q$, and the decryption key is $-x$. The algorithm Enc on input an encryption key ek_i and a message $m \in \mathcal{M}$, samples $r_1, r_2 \in \mathbb{Z}_q$ uniformly at random and outputs the ciphertext

$$c = (c_0, c_1, c_2, c_3) \coloneqq (g^{r_1}, h^{r_1} g^{ek_i}, g^{r_2}, m \cdot h^{r_2}).$$

We can define the function $\mathsf{maul}_{\mathsf{DHO}}$ as

$$\mathsf{maul}_{\mathsf{DHO}}\big((c_0, c_1, c_2, c_3), sp, m, m'\big) \coloneqq \big(c_0, c_1, c_2, m' \cdot m^{-1} \cdot c_3\big).$$

Since the group order q is part of sp, this function is efficiently computable. For $c_3 = m \cdot h^{r_2}$, we thus get a new fourth component $c_3' = m' \cdot h^{r_2}$ and equation (1) is satisfied.

The malleability for more than one identity (and in particular in our scenario described above) follows since the scheme for several identities is composed of independent instances of the basic single-identity scheme.

3.3 FGKO Scheme Based on ElGamal

Description of the scheme. In that scheme, the sanitizer parameters consist of the description of a finite cyclic group $G = \langle g \rangle$ including the group order q and a generator g, a verification key vk^{Sig} of a signature scheme Sig, and a common-reference string crs^{NIZK} of a NIZK proof system NIZK for the language $L := \{x \mid \exists w\ (x,w) \in R\}$, where R is defined as follows: for $x = \big(vk^{\mathsf{Sig}}, c_0, c_1, c_2, c_3\big)$ and a witness $w = \big(g^x, \sigma^{\mathsf{Sig}}, m, r, s\big)$, $R(x,w) = 1$ if and only if

$$\mathsf{Sig.Ver}\big(vk^{\mathsf{Sig}}, g^x, \sigma^{\mathsf{Sig}}\big) = 1 \ \wedge\ (c_0, c_1, c_2, c_3) = \big(g^r, g^{x\cdot r}, g^s, m \cdot g^{x\cdot s}\big).$$

The encryption and decryption keys are given by $ek := (g^x, \sigma^{\mathsf{Sig}})$, $dk := x$ for a uniformly chosen $x \twoheadleftarrow \mathbb{Z}_q$, where σ^{Sig} is a signature on g^x. To encrypt a message m, first choose $r \twoheadleftarrow \mathbb{Z}_q^*$ and $s \twoheadleftarrow \mathbb{Z}_q$ uniformly at random and compute $(c_0, c_1, c_2, c_3) := (g^r, g^{x\cdot r}, g^s, m \cdot g^{x\cdot s})$. Then run $\pi^{\mathsf{NIZK}} \leftarrow \mathsf{NIZK.Prove}\big(crs^{\mathsf{NIZK}}, (vk^{\mathsf{Sig}}, c_0, c_1, c_2, c_3), (g^x, \sigma^{\mathsf{Sig}}, m, r, s)\big)$ and output the ciphertext $c := (c_0, c_1, c_2, c_3, \pi)$.

Potential malleability. We define the function $\mathsf{maul}_{\mathsf{FGKO}}$ as

$$\mathsf{maul}_{\mathsf{FGKO}}\big((c_0, c_1, c_2, c_3, \pi), sp, m, m'\big) := \big(c_0, c_1, c_2, m' \cdot m^{-1} \cdot c_3, \pi\big).$$

This function satisfies equation (1) if, for example, the non-interactive zero-knowledge proof is independent of the last component c_3. We show that such a NIZK proof system exists without violating the properties assumed by Fuchsbauer et al. [FGKO17]. To this end, let NIZK' be a NIZK proof system for the language $L' := \{x \mid \exists w\ (x,w) \in R'\}$, where the relation R' is defined as follows: for $x = \big(vk^{\mathsf{Sig}}, c_0, c_1, c_2\big)$ and $w = \big(g^x, \sigma^{\mathsf{Sig}}, r, s\big)$, $(x,w) \in R'$ if and only if

$$\mathsf{Sig.Ver}\big(vk^{\mathsf{Sig}}, g^x, \sigma^{\mathsf{Sig}}\big) = 1 \ \wedge\ (c_0, c_1, c_2) = \big(g^r, g^{x\cdot r}, g^s\big).$$

Given NIZK', we construct a NIZK proof system NIZK for the original language L as follows:

$$\begin{aligned}
&\mathsf{NIZK.Gen}(1^\kappa) := \mathsf{NIZK}'.\mathsf{Gen}(1^\kappa),\\
&\mathsf{NIZK.Prove}\big(crs^{\mathsf{NIZK}}, (vk^{\mathsf{Sig}}, c_0, c_1, c_2, c_3), (g^x, \sigma^{\mathsf{Sig}}, m, r, s)\big) :=\\
&\qquad\qquad \mathsf{NIZK}'.\mathsf{Prove}\big(crs^{\mathsf{NIZK}}, (vk^{\mathsf{Sig}}, c_0, c_1, c_2), (g^x, \sigma^{\mathsf{Sig}}, r, s)\big),\\
&\mathsf{NIZK.Ver}\big(crs^{\mathsf{NIZK}}, (vk^{\mathsf{Sig}}, c_0, c_1, c_2, c_3), \pi^{\mathsf{NIZK}}\big) := \mathsf{NIZK}'.\mathsf{Ver}\big(crs^{\mathsf{NIZK}}, (vk^{\mathsf{Sig}}, c_0, c_1, c_2), \pi^{\mathsf{NIZK}}\big).
\end{aligned}$$

Correctness and zero-knowledge of NIZK follow straightforwardly from the underlying scheme NIZK'. For knowledge-extraction, assume that NIZK' is capable of extracting a valid witness $(g^x, \sigma^{\mathsf{Sig}}, r, s)$ given a valid proof for the statement $(vk^{\mathsf{Sig}}, c_0, c_1, c_2)$. Given a statement $\big(vk^{\mathsf{Sig}}, c_0, c_1, c_2, c_3\big)$ in the original language L, we can obtain a valid message encoded in c_3 by computing $m := c_3 \cdot (g^{x\cdot s})^{-1}$, and thus also a witness $\big(g^x, \sigma^{\mathsf{Sig}}, m, r, s\big)$ for the given statement. Finally, for soundness, note that if $(vk^{\mathsf{Sig}}, c_0, c_1, c_2) \in L'$, this implies that any group element $c_3 \in G$ is a valid last component, i.e., $(vk^{\mathsf{Sig}}, c_0, c_1, c_2, c_3) \in L$ for any $c_3 \in G$, since there exists the message $m := c_3 \cdot (g^{x\cdot s})^{-1}$, and thus a valid witness $w = (g^x, \sigma^{\mathsf{Sig}}, m, r, s)$.

For the constructed scheme NIZK and the function $\mathsf{maul}_{\mathsf{FGKO}}$, equation (1) clearly holds. Hence, the FGKO scheme can be instantiated such that the malleability attack works. It could potentially be excluded by requiring stronger properties from the NIZK scheme.

4 A Stronger Notion of ACE

In this section, we introduce our new security definitions, which exclude the issues we have discovered.

4.1 ACE with Modification Detection

To be resilient against the ciphertext-revealing attacks described in Section 3, the sanitizer should ideally only sanitize fresh encryptions and block ciphertexts that are either replays or obtained by modifying previous ciphertexts. Therefore, we introduce an additional algorithm for detecting modified ciphertexts. If the sanitizer receives a ciphertext that is detected to be a modification of a previously received one, this ciphertext is blocked. Since such ciphertexts will not be stored in the repository and consequently not be decrypted, we define chosen-ciphertext security with respect to a decryption oracle that does not return a decryption if the received ciphertext is detected to be a modification of the challenge ciphertext. Our definitions can therefore be seen as a variant of publicly-detectable replayable-CCA security, which was introduced by Canetti et al. [CKN03] for public key encryption. Before defining the security, we define the syntax of ACE schemes with this additional algorithm.

Definition 4.1. An *access control encryption with modification detection scheme* is an ACE scheme $\mathcal{E}$ together with a PPT algorithm DMod that on input sanitizer parameters sp and two ciphertexts $c, \tilde{c} \in \mathcal{C}$, outputs a bit b (where $b = 1$ means that $\tilde{c}$ was obtained via modifying c).

Except for Section 4.3, where we show that our new definitions imply the existing ones, we will from now on only consider ACE schemes with modification detection and thus often refer to them simply as ACE schemes.

The algorithm DMod should output 1 if $\tilde{c}$ is an adversarial modification of c, and 0 otherwise. We have the following intuitive requirements on DMod:

1. All ciphertexts $\tilde{c}$ an adversary can produce given ciphertexts $c_1, \ldots, c_l$ and no encryption key, are either invalid (i.e., sanitize to $\perp$) or we have $\mathsf{DMod}(sp, c_i, \tilde{c}) = 1$ for some $i \in \{1, \ldots, n\}$.

2. Given encryption and decryption keys, an adversary is unable to produce a ciphertext c such that a ciphertext produced by Enc for a message of the adversary's choice is detected to be a modification of c. In particular, independent encryptions of messages collide only with negligible probability.

The first requirement is captured by role-respecting security as defined in Definition 4.5, the second one by non-detection of fresh encryptions defined in Definition 4.4.

Remark. Canetti et al. (translated to our setting) also require that if $\mathsf{DMod}(sp, c, \tilde{c}) = 1$, then c and $\tilde{c}$ decrypt to the same message [CKN03]. For our purpose, this is not needed. This means that we do not want to detect replays in the sense that the same message is replayed, but more generally, whether the given ciphertext was obtain via some modification of another ciphertext.

4.2 New Security Definitions

We formalize chosen-ciphertext attacks by giving the adversary access to an oracle $\mathcal{O}_{SD}$ that first sanitizes a given ciphertext and then decrypts the result. One could also consider *chosen-sanitized-ciphertext attacks* by providing the adversary access to an oracle $\mathcal{O}_D$ that only decrypts.

This is potentially stronger since the adversary can emulate the oracle $\mathcal{O}_{SD}$ by first sanitizing the ciphertexts and then giving the result to $\mathcal{O}_D$, but given $\mathcal{O}_{SD}$, it is not necessarily possible to emulate $\mathcal{O}_D$. Since in the application, users can only send ciphertexts to the sanitizer but not directly write ciphertexts to the repository such that they are decrypted without being sanitized, the weaker notion is sufficient.

In principle, the adversary has in all definitions access to $\mathcal{O}_{SD}$, as well as to an encryption oracle and a key-generation oracle. To simplify the definitions, we omit the encryption or decryption oracles if the winning condition places no restriction on the encryption or decryption keys obtained from the key-generation oracle, respectively.

Privacy and anonymity. We now define (payload) privacy and sender-anonymity. The former guarantees that encryptions of different messages under the same encryption key cannot be distinguished as long as the adversary has no decryption key that allows to decrypt. We also require this for messages of different length, i.e., schemes satisfying our definition do not leak the length of the encrypted message, which means that the message space has to be bounded. Anonymity guarantees that encryptions of the same message under different keys cannot be distinguished. We distinguish a weak and a strong variant of anonymity, where the weak one provides no guarantees if the adversary can decrypt the ciphertext, and the strong one guarantees that even if the adversary has decryption keys, nothing is leaked about the sender role beyond which of the adversary's decryption keys can be used to decrypt.

Definition 4.2. Let $\mathcal{E} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{San}, \mathsf{Dec}, \mathsf{DMod})$, be an ACE with modification detection scheme and let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a pair of probabilistic algorithms. Consider the experiment $\mathsf{Exp}^{\mathsf{ACE\text{-}PRV\text{-}ANON\text{-}CCA}}_{\mathcal{E},\mathcal{A}}$ in Figure 2 and let J be the set of all j such that $\mathcal{A}_1$ or $\mathcal{A}_2$ issued the query $(j, \mathtt{rec})$ to the oracle $\mathcal{O}_G$. We define the *privacy under chosen-ciphertext attacks advantage* and the *sender-anonymity under chosen-ciphertext attacks advantages* of $\mathcal{A}$ as

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{ACE\text{-}PRV\text{-}CCA}}_{\mathcal{E},\mathcal{A}} &:= 2 \cdot \Pr\big[b' = b \ \wedge \ i_0 = i_1 \ \wedge \ \forall j \in J \ P(i_0, j) = 0\big] - 1,\\
\mathsf{Adv}^{\mathsf{ACE\text{-}wANON\text{-}CCA}}_{\mathcal{E},\mathcal{A}} &:= 2 \cdot \Pr\big[b' = b \ \wedge \ m_0 = m_1 \ \wedge \ \forall j \in J \ P(i_0, j) = P(i_1, j) = 0\big] - 1,\\
\mathsf{Adv}^{\mathsf{ACE\text{-}sANON\text{-}CCA}}_{\mathcal{E},\mathcal{A}} &:= 2 \cdot \Pr\big[b' = b \ \wedge \ m_0 = m_1 \ \wedge \ \forall j \in J \ P(i_0, j) = P(i_1, j)\big] - 1,
\end{aligned}$$

respectively, where all probabilities are in $\mathsf{Exp}^{\mathsf{ACE\text{-}PRV\text{-}ANON\text{-}CCA}}_{\mathcal{E},\mathcal{A}}$. The scheme $\mathcal{E}$ is called *private under chosen-ciphertext attacks (PRV-CCA secure)*, *weakly sender-anonymous under chosen-ciphertext attacks (wANON-CCA secure)*, and *strongly sender-anonymous under chosen-ciphertext attacks (sANON-CCA secure)* if $\mathsf{Adv}^{\mathsf{ACE\text{-}PRV\text{-}CCA}}_{\mathcal{E},\mathcal{A}}$, $\mathsf{Adv}^{\mathsf{ACE\text{-}wANON\text{-}CCA}}_{\mathcal{E},\mathcal{A}}$, and $\mathsf{Adv}^{\mathsf{ACE\text{-}sANON\text{-}CCA}}_{\mathcal{E},\mathcal{A}}$ are negligible for all efficient $\mathcal{A}$, respectively.

Remark. Weak anonymity corresponds to the anonymity notion considered by Fuchsbauer et al. [FGKO17] and strong anonymity to the one considered by Damgård et al. [DHO16]. We state both definitions because weak anonymity is easier to achieve but strong anonymity might be required by some applications. If anonymity is only required against the sanitizer or if all messages are anyway signed by the sender, weak anonymity is sufficient. Strong anonymity is required in settings where senders also want to retain as much anonymity as possible against legitimate receivers.

Sanitization security. We next define sanitization security, which excludes that dishonest parties can communicate via the ciphertexts. We formalize this by requiring that the output

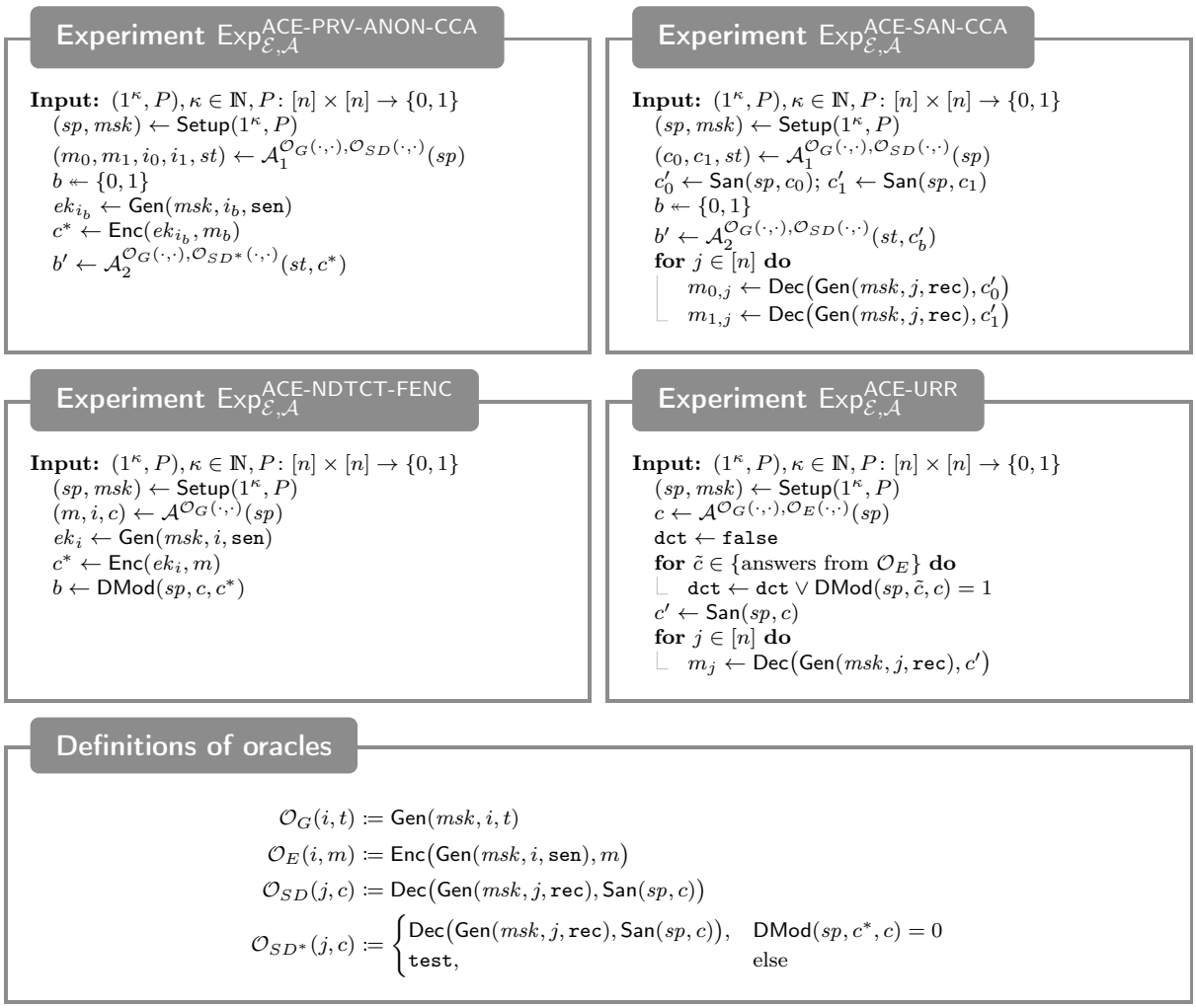

Figure 2: Security experiments for an ACE with modification detection scheme $\mathcal{E}$ and an adversary $\mathcal{A}$, where $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ in the first two experiments.

of the sanitizer for two different ciphertexts cannot be distinguished, as long as both sanitized ciphertexts are not $\perp$ and the adversary has no decryption key that decrypts one of the ciphertexts. This provides no security guarantees if the adversary can decrypt the ciphertexts, which does not seem to be an issue since in this case, the parties can anyway directly communicate via the messages. However, we additionally consider a stronger variant, where the adversary is allowed to possess a decryption key that decrypts the ciphertexts, as long as they both decrypt to the same message. This stronger variant excludes subliminal channels, i.e., even if the involved parties are allowed to communicated by the policy, they cannot exchange information via ciphertexts beyond the encrypted message.

Since the adversary provides the two ciphertexts that are sanitized, we do not know to which roles they correspond; they could even be particularly crafted without belonging to an existing role. Hence, we cannot state the requirement (in the weak variant) that the adversary should not be able to decrypt by only considering the policy and the obtained decryption keys, as in the

no-write rule in Definition 2.4. Instead, we require that the decryption algorithm returns $\perp$ for all decryption keys the adversary possesses. For this to provide the intended security, we need that the decrypt algorithm returns $\perp$ whenever the receiver role corresponding to the used key is not supposed to read the message. This is guaranteed by role-respecting security which is defined later.

Definition 4.3. Let $\mathcal{E} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{San}, \mathsf{Dec}, \mathsf{DMod})$ be an ACE with modification detection scheme and let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a pair of probabilistic algorithms. Consider the experiment $\mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}}$ in Figure 2 and let J be the set of all j such that $\mathcal{A}_1$ or $\mathcal{A}_2$ issued the query $(j, \mathtt{rec})$ to the oracle $\mathcal{O}_G$. We define the *sanitization under chosen-ciphertext attacks advantage* and the *strong sanitization under chosen-ciphertext attacks advantage* of $\mathcal{A}$ as

$$\mathsf{Adv}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}} := 2 \cdot \Pr\big[b' = b \ \wedge \ c'_0 \neq \perp \neq c'_1 \ \wedge \ \forall j \in J \ m_{0,j} = m_{1,j} = \perp\big] - 1,$$
$$\mathsf{Adv}^{\mathsf{ACE\text{-}sSAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}} := 2 \cdot \Pr\big[b' = b \ \wedge \ c'_0 \neq \perp \neq c'_1 \ \wedge \ \forall j \in J \ m_{0,j} = m_{1,j}\big] - 1,$$

respectively, where the probability is over the randomness in $\mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}}$. The scheme $\mathcal{E}$ is called *sanitization under chosen-ciphertext attacks secure (SAN-CCA secure)* and *strongly sanitization under chosen-ciphertext attacks secure (sSAN-CCA secure)* if $\mathsf{Adv}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}}$ and $\mathsf{Adv}^{\mathsf{ACE\text{-}sSAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}}$ are negligible for all efficient $\mathcal{A}$, respectively.

Non-detection of fresh encryptions. In the intended way of using a scheme satisfying our notions, the sanitizer only adds sanitized ciphertexts to the repository if the given ciphertext is not detected to be a modification of a previously received ciphertext. This means that if an adversary can find a ciphertext c such that another ciphertext c^* that is later honestly generated is detected as a modification of c, the delivery of the message at that later point can be prevented by sending the ciphertext c to the sanitizer earlier. We exclude this by the following definition, which can be seen as an extended correctness requirement.

Definition 4.4. Let $\mathcal{E} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{San}, \mathsf{Dec}, \mathsf{DMod})$ be an ACE with modification detection scheme and let $\mathcal{A}$ be a probabilistic algorithm. Consider the experiment $\mathsf{Exp}^{\mathsf{ACE\text{-}NDTCT\text{-}FENC}}_{\mathcal{E},\mathcal{A}}$ in Figure 2. We define the *non-detection of fresh encryptions advantage* of $\mathcal{A}$ as

$$\mathsf{Adv}^{\mathsf{ACE\text{-}NDTCT\text{-}FENC}}_{\mathcal{E},\mathcal{A}} := \Pr\big[b = 1\big],$$

where the probability is over the randomness in $\mathsf{Exp}^{\mathsf{ACE\text{-}NDTCT\text{-}FENC}}_{\mathcal{E},\mathcal{A}}$. The scheme $\mathcal{E}$ is said to have *non-detecting fresh encryptions (NDTCT-FENC)* if $\mathsf{Adv}^{\mathsf{ACE\text{-}NDTCT\text{-}FENC}}_{\mathcal{E},\mathcal{A}}$ is negligible for all efficient $\mathcal{A}$.

Role-respecting and uniform-decryption security. We finally define role-respecting and uniform-decryption security. The former means that an adversary cannot produce a ciphertext for which the pattern of roles that can decrypt does not correspond to a role for which the adversary has an encryption key. For example, if the adversary has only an encryption key for the role i such that roles j_0 and j_1 are the only roles j with $P(i, j) = 1$, all ciphertexts produced by the adversary are either invalid (i.e., sanitized to $\perp$ or detected as a modification) or decrypt to a message different from $\perp$ precisely under the decryption keys for j_0 and j_1. On the one hand, this means that receivers who are not allowed to receive the message get $\perp$ and hence

know that the message is not for them.[2] On the other hand, it also guarantees that the adversary cannot prevent receivers with role j_1 from receiving a message that is sent to receivers with role j_0. Furthermore, uniform decryption guarantees for all ciphertexts c output by an adversary that if c decrypts to a message different from $\perp$ for different decryption keys, it always decrypts to the same message. In the example above, this means that j_0 and j_1 not only both receive *some* message, but they both receive *the same* one.

Definition 4.5. Let $\mathcal{E} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{San}, \mathsf{Dec}, \mathsf{DMod})$ be an ACE with modification detection scheme and let $\mathcal{A}$ be a probabilistic algorithm. Consider the experiment $\mathsf{Exp}^{\mathsf{ACE\text{-}URR}}_{\mathcal{E},\mathcal{A}}$ in Figure 2 and let I and J be the sets of all i and j such that $\mathcal{A}$ issued the query $(i, \mathtt{sen})$ and $(j, \mathtt{rec})$ to the oracle $\mathcal{O}_G$, respectively. We define the *role-respecting advantage* and the *uniform-decryption advantage* of $\mathcal{A}$ as

$$\mathsf{Adv}^{\mathsf{ACE\text{-}RR}}_{\mathcal{E},\mathcal{A}} := \Pr\big[c' \neq \perp \ \wedge\ \mathtt{dct} = \mathtt{false} \ \wedge\ \neg\big(\exists i \in I\ \forall j \in J\ (m_j \neq \perp \leftrightarrow P(i,j) = 1)\big)\big],$$
$$\mathsf{Adv}^{\mathsf{ACE\text{-}UDEC}}_{\mathcal{E},\mathcal{A}} := \Pr\big[\exists j, j' \in J\ m_j \neq \perp \neq m_{j'} \ \wedge\ m_j \neq\ m_{j'}\big],$$

respectively, where the probabilities are over the randomness in $\mathsf{Exp}^{\mathsf{ACE\text{-}URR}}_{\mathcal{E},\mathcal{A}}$. The scheme $\mathcal{E}$ is *role-respecting (RR secure)* and *uniform-decryption (UDEC) secure* if $\mathsf{Adv}^{\mathsf{ACE\text{-}RR}}_{\mathcal{E},\mathcal{A}}$ and $\mathsf{Adv}^{\mathsf{ACE\text{-}UDEC}}_{\mathcal{E},\mathcal{A}}$ are negligible for all efficient $\mathcal{A}$, respectively.

Remark. Note that in Definition 4.5, we only check the decryptions for receiver roles for which $\mathcal{A}$ has requested the corresponding decryption key. This means that an adversary in addition to producing a ciphertext that causes an inconsistency, also has to find a receiver role for which this inconsistency manifests. If the total number of roles n is small (say polynomial in the security parameter), $\mathcal{A}$ can simply query $\mathcal{O}_G$ on all receiver keys, but for large n this condition is nontrivial. For example, we consider a scheme secure if an adversary can efficiently produce a ciphertext such that there is a receiver role that can decrypt it even though the policy does not allow it, as long as this receiver role is hard to find. The rationale is that in this case, the inconsistency cannot be exploited and will only be observed with negligible probability in an execution of the protocol.

4.3 Relation to the Original Security Notions

In this section, we discuss how our notions relate to the original security definitions (see Section 2.4). First note that we assume the scheme has an additional algorithm DMod. As explained in Section 4.1, the intended usage of such a scheme is that the sanitizer discards ciphertexts that are detected to be a modification of a previous ciphertext. This means that if dishonest parties want to communicate even though disallowed by the policy (i.e., they want to break the no-write rule), the sender must produce a ciphertext that is not detected as a modification of a previous ciphertext. With this in mind, it is natural to adjust the no-write rule such that an adversary only wins if the ciphertext he outputs is not detected to be a modification of a ciphertext generated by the oracle $\mathcal{O}_{ES}$ (before sanitizing it).

[2] Detectability (Definition 2.2) provides this guarantee for honest encryptions, role-respecting security extends this to maliciously generated ciphertexts. Note, however, that detectability is not implied by role-respecting security: If an adversary has encryption keys for two roles i and i', role-respecting security does not exclude that encrypting some message (depending on i') with the key for role i can be decrypted with keys for roles that are allowed to receive from i'.

Definition 4.6. Let Let $\mathcal{E} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{San}, \mathsf{Dec}, \mathsf{DMod})$ be an ACE with modification detection scheme and let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a pair of probabilistic algorithms. The experiment $\mathsf{Exp}^{\text{ACE-MD-no-write}}_{\mathcal{E},\mathcal{A}}$ is identical to $\mathsf{Exp}^{\text{ACE-no-write}}_{\mathcal{E},\mathcal{A}}$ in Figure 1 except that after $\mathcal{A}_1$ returns (c_0, i', st), it is checked whether the oracle $\mathcal{O}_{ES}$ has generated some $\tilde{c}$ and returned its sanitization such that $\mathsf{DMod}(sp, \tilde{c}, c_0) = 1$. If this is the case, set $\texttt{dct} \leftarrow \texttt{true}$, else $\texttt{dct} \leftarrow \texttt{false}$. Let I_1 be the set of all i such that $\mathcal{A}_1$ issued the query $(i, \texttt{sen})$ to $\mathcal{O}_G$, and let J be the set of all j such that $\mathcal{A}_1$ or $\mathcal{A}_2$ issued the query $(j, \texttt{rec})$ to $\mathcal{O}_G$. We define the *no-write with modification detection advantage* of $\mathcal{A}$ as

$$\mathsf{Adv}^{\text{ACE-MD-no-write}}_{\mathcal{E},\mathcal{A}} := 2 \cdot \Pr\big[b' = b \ \wedge \ \texttt{dct} = \texttt{false} \ \wedge \ i' \in I_1 \\ \wedge \ \forall i \in I_1 \ \forall j \in J \ P(i,j) = 0 \ \wedge \ \mathsf{San}(sp, c_0) \neq \bot\big] - 1,$$

where the probability is over the randomness of all algorithms in $\mathsf{Exp}^{\text{ACE-MD-no-write}}_{\mathcal{E},\mathcal{A}}$. The scheme $\mathcal{E}$ satisfies the *no-write with modification detection rule* if $\mathsf{Adv}^{\text{ACE-MD-no-write}}_{\mathcal{E},\mathcal{A}}$ is negligible for all efficient $\mathcal{A}$.

We show that our new security definitions from Section 4.2 imply the no-read rule and the no-write with modification detection rule. We have to assume that the policy P allows for all i that one can efficiently find some j with $P(i,j) = 1$. This seems to be the case for all practically relevant policies, though. The results are summarized in the following theorem.

Theorem 4.7. *Let $\mathcal{E} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{San}, \mathsf{Dec}, \mathsf{DMod})$ be an ACE with modification detection scheme and let $\mathcal{E}' = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{San}, \mathsf{Dec})$ be the corresponding ACE scheme. If $\mathcal{E}$ is correct and PRV-CCA, sANON-CCA, SAN-CCA, and RR secure, then it satisfies the the no-write with modification detection rule for policies P such that for all i, one can efficiently find some j with $P(i,j) = 1$, and $\mathcal{E}'$ satisfies the no-read rule. More precisely, for all adversaries $\mathcal{A}$, $\mathcal{A}'$, and $\mathcal{A}''$, there exist adversaries $\mathcal{A}_{\mathsf{PRV}}$ and $\mathcal{A}_{\mathsf{wANON}}$ (both roughly as efficient as emulating an execution of $\mathsf{Exp}^{\text{ACE-no-read}}_{\mathcal{E}',\mathcal{A}}$), an adversary $\mathcal{A}'_{\mathsf{sANON}}$ (roughly as efficient as emulating an execution of $\mathsf{Exp}^{\text{ACE-no-read}}_{\mathcal{E}',\mathcal{A}'}$), and adversaries $\mathcal{A}''_{\mathsf{SAN}}$, $\mathcal{A}''_{\mathsf{RR}}$, and $\mathcal{A}''_{\mathsf{CORR}}$ (all roughly as efficient as emulating an execution of $\mathsf{Exp}^{\text{ACE-MD-no-write}}_{\mathcal{E},\mathcal{A}''}$) such that*

$$\begin{aligned}
\mathsf{Adv}^{\text{ACE-no-read,priv}}_{\mathcal{E}',\mathcal{A}} &\leq \mathsf{Adv}^{\text{ACE-PRV-CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{PRV}}} + \mathsf{Adv}^{\text{ACE-wANON-CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{wANON}}}, \\
\mathsf{Adv}^{\text{ACE-no-read,anon}}_{\mathcal{E}',\mathcal{A}'} &= \mathsf{Adv}^{\text{ACE-sANON-CCA}}_{\mathcal{E},\mathcal{A}'_{\mathsf{sANON}}}, \\
\mathsf{Adv}^{\text{ACE-MD-no-write}}_{\mathcal{E},\mathcal{A}''} &\leq \mathsf{Adv}^{\text{ACE-SAN-CCA}}_{\mathcal{E},\mathcal{A}''_{\mathsf{SAN}}} + 4 \cdot \mathsf{Adv}^{\text{ACE-RR}}_{\mathcal{E},\mathcal{A}''_{\mathsf{RR}}} + 2 \cdot \mathsf{Adv}^{\text{ACE-CORR}}_{\mathcal{E},\mathcal{A}''_{\mathsf{CORR}}}.
\end{aligned}$$

We here sketch the proof idea, a detailed proof of the theorem is provided in Appendix B. To prove the claim about the payload-privacy no-read rule, consider the hybrid experiment H that is identical to $\mathsf{Exp}^{\text{ACE-no-read}}_{\mathcal{E}',\mathcal{A}}$ except that after $\mathcal{A}_1$ returns (m_0, m_1, i_0, i_1, st), i_1 is replaced by i_0. If $\mathcal{A}$ wins the no-read privacy game, $P(i_0, j) = P(i_1, j) = 0$ for all j for which $\mathcal{A}$ obtained a decryption key. Hence, in this case $\mathsf{Exp}^{\text{ACE-no-read}}_{\mathcal{E}',\mathcal{A}}$ and H are indistinguishable by weak sender-anonymity. If $\mathcal{A}$ wins in H, one can construct an adversary against PRV-CCA security by running $\mathcal{A}$, returning (m_0, m_1, i_0, i_0, st) when $\mathcal{A}_1$ returns (m_0, m_1, i_0, i_1, st), and returning the same guess as $\mathcal{A}_2$. Note that $\mathcal{A}$ has access to an encryption oracle $\mathcal{O}_E$ in $\mathsf{Exp}^{\text{ACE-no-read}}_{\mathcal{E}',\mathcal{A}}$, which is not available in $\mathsf{Exp}^{\text{ACE-PRV-ANON-CCA}}_{\mathcal{E},\mathcal{A}}$. However, since the winning conditions do not restrict the encryption keys obtained from $\mathcal{O}_G$, the oracle $\mathcal{O}_E$ can be emulated by obtaining the encryption key and then encrypting the message.

Relating the sender-anonymity no-read rule to sANON-CCA security is a straightforward reduction.

To proof the claim about the no-write rule, assume $\mathcal{A}''$ wins the corresponding game. If $\mathcal{A}''$ does not obtain a decryption key that decrypts c_0 or c_1 to a message different from $\perp$, this adversary can be used to break SAN-CCA security as follows: when $\mathcal{A}_1''$ returns (c_0, i', st), output c_0 and the encryption of a uniformly chosen message for sender role i' as c_1; finally output the same guess b' as $\mathcal{A}_2''$. Correctness ensures that c_1 does not sanitize to $\perp$,[3] so the winning condition of the SAN-CCA game is satisfied. If $\mathcal{A}''$ does obtain a decryption key that decrypts c_0 or c_1 to a message different from $\perp$, one can construct an adversary against role-respecting security.

Relation to original no-write rule. Perhaps surprisingly, one can also show that our new notions imply the original no-write rule if DMod is symmetric in the sense that $\Pr[\mathsf{DMod}(sp, c_0, c_1) = 1] = \Pr[\mathsf{DMod}(sp, c_1, c_0) = 1]$ (which is the case for all schemes considered in this paper). The proof idea is to construct adversaries against correctness, and sanitization and role-respecting security as above. Now, the role-respecting game is not won if the adversary $\mathcal{A}$ returns a ciphertext c_0 that is detected to be a modification of a ciphertext generated by $\mathcal{O}_{ES}$. We show that in this case, we can break sSAN-CCA security. Note that $\mathcal{O}_{ES}$ only gives $\mathcal{A}$ the sanitized ciphertexts. The proof idea is as follows. $\mathcal{A}_1$ makes several queries to $\mathcal{O}_{ES}$. For a uniformly chosen one, encrypt the message twice, give the resulting ciphertexts $\tilde{c}_0, \tilde{c}_1$ to the sSAN-CCA challenger, and give the obtained sanitized ciphertext $\tilde{c}_b'$ to $\mathcal{A}_1$. For all other queries, encrypt and sanitize the message normally. When $\mathcal{A}_1$ returns a ciphertext c_0, check whether c_0 is detected to be a modification of $\tilde{c}_0$ or $\tilde{c}_1$. Since the ciphertext $\tilde{c}_{1-b}$ is information-theoretically hidden from $\mathcal{A}$, it can be considered to be a fresh encryption. By our assumption, the probability that c_0 is detected to be a modification of $\tilde{c}_{1-b}$ is equal to the probability that $\tilde{c}_{1-b}$ is detected to be a modification of c_0, which contradicts non-detection of fresh encryptions. Hence, by checking which of the two ciphertexts is detected, one can guess b and thus break sSAN-CCA security. Note that $\mathcal{A}$ is allowed to obtain a decryption key that decrypts $\tilde{c}_b'$, which is why we need *strong* sanitization security.

Theorem 4.8. *Let $\mathcal{E} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{San}, \mathsf{Dec}, \mathsf{DMod})$ be an ACE with modification detection scheme such that $\Pr[\mathsf{DMod}(sp, c_0, c_1) = 1] = \Pr[\mathsf{DMod}(sp, c_1, c_0) = 1]$ for all sp returned by Setup and all ciphertexts $c_0, c_1 \in \mathcal{C}$. Further let $\mathcal{E}' = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{San}, \mathsf{Dec})$ be the corresponding ACE scheme. If $\mathcal{E}$ is correct, detectable, has NDTCT-FENC, and is sSAN-CCA and RR secure, then $\mathcal{E}'$ satisfies the no-write rule for policies P such that for all i, one can efficiently find some j with $P(i, j) = 1$. More precisely, for all adversaries $\mathcal{A}$ that make at most q_{ES} queries to the oracle $\mathcal{O}_{ES}$ and at most q_{dk} queries of the form $(\cdot, \mathtt{rec})$ to $\mathcal{O}_G$, there exist adversaries $\mathcal{A}_{\mathsf{SAN}}$, $\mathcal{A}_{\mathsf{RR}}$, $\mathcal{A}_{\mathsf{sSAN}}$, $\mathcal{A}_{\mathsf{NDTCT}}$, $\mathcal{A}_{\mathsf{CORR}}$, and $\mathcal{A}_{\mathsf{dtct}}$ (all roughly as efficient as emulating an execution of $\mathsf{Exp}^{\mathsf{ACE\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}}$) such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{ACE\text{-}no\text{-}write}}_{\mathcal{E}',\mathcal{A}} &\leq \mathsf{Adv}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{SAN}}} + 4 \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}RR}}_{\mathcal{E},\mathcal{A}_{\mathsf{RR}}} + 2q_{ES} \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}sSAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{sSAN}}} \\ &\quad + 4q_{ES} \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}NDTCT\text{-}FENC}}_{\mathcal{E},\mathcal{A}_{\mathsf{NDTCT}}} + (8q_{ES}q_{dk} + 2) \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}CORR}}_{\mathcal{E},\mathcal{A}_{\mathsf{CORR}}} + 8q_{ES}q_{dk} \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}DTCT}}_{\mathcal{E},\mathcal{A}_{\mathsf{dtct}}}.\end{aligned}$$

See Appendix B for a detailed proof.

[3] This is the only place where we need that one can efficiently find j with $P(i', j) = 1$ since the adversary in the correctness game has to provide such j.

5 Enhanced Sanitizable Public-Key Encryption

5.1 Definitions

As a stepping stone toward ACE schemes satisfying our new security definitions, we introduce *enhanced sanitizable public-key encryption*. Sanitizable public-key encryption has been considered by Damgård et al. [DHO16] and Fuchsbauer et al. [FGKO17] as a relaxation of universal re-encryption [GJJS04] and rerandomizable encryption [Gro04; PR07]. It allows to *sanitize* a ciphertext to obtain a *sanitized ciphertext* that cannot be linked to the original ciphertext except that it decrypts to the correct message. In contrast to rerandomizable encryption, sanitized ciphertexts can have a different syntax than ciphertexts, i.e., it is not required that a sanitized ciphertext is indistinguishable from a fresh encryption. We introduce an enhanced variant with a different syntax and stronger security guarantees.

Definition 5.1. An *enhanced sanitizable public-key encryption (sPKE) scheme* consists of the following five PPT algorithms:

Setup: The algorithm Setup on input a security parameter 1^κ, outputs *sanitizer parameters* sp, and a *master secret key* msk. We implicitly assume that all parameters and keys include the finite *message space* $\mathcal{M}$ and the *ciphertext spaces* $\mathcal{C}, \mathcal{C}'$.

Key generation: The algorithm Gen on input a master secret key msk, outputs an *encryption key* ek and a *decryption key* dk.

Encryption: The algorithm Enc on input an encryption key ek and a message $m \in \mathcal{M}$, outputs a ciphertext $c \in \mathcal{C}$.

Sanitization: The algorithm San on input sanitizer parameters sp and a ciphertext $c \in \mathcal{C}$, outputs a *sanitized ciphertext* $c' \in \mathcal{C}' \cup \{\bot\}$.

Decryption: The algorithm Dec on input a decryption key dk and a sanitized ciphertext $c' \in \mathcal{C}'$, outputs a message $m \in \mathcal{M} \cup \{\bot\}$; on input dk and $\bot$, it outputs $\bot$.

For correctness, we require for all (sp, msk) in the range of Setup, all (ek, dk) in the range of $\mathsf{Gen}(msk)$, and all $m \in \mathcal{M}$ that

$$\mathsf{Dec}\big(dk, \mathsf{San}\big(sp, \mathsf{Enc}(ek, m)\big)\big) = m$$

with probability 1.

We require robustness in the sense that no ciphertext decrypts to a message different from $\bot$ for two different decryption keys (except with negligible probability). This is similar to detectability for ACE schemes, but we allow the adversary to directly output a ciphertext, instead of a message, which is then honestly encrypted. Our notion therefore closely resembles *unrestricted strong robustness (USROB)*, introduced by Farshim et al. [FLPQ13] for public-key encryption, which also allows the adversary to choose a ciphertext and, in contrast to strong robustness by Abdalla et al. [ABN10], gives the adversary access to decryption keys.

Definition 5.2. Let $\mathcal{E} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{San}, \mathsf{Dec})$ be an sPKE scheme. For a probabilistic algorithm $\mathcal{A}$, we define the experiment $\mathsf{Exp}^{\mathsf{sPKE\text{-}USROB}}_{\mathcal{E},\mathcal{A}}$ that executes $(sp, msk) \leftarrow \mathsf{Setup}(1^\kappa)$ and $(c, i_0, i_1) \leftarrow \mathcal{A}^{\mathcal{O}_G(\cdot)}(sp)$, where the oracle $\mathcal{O}_G$ on input `getNew`, outputs a fresh key pair

$(ek, dk) \leftarrow \mathsf{Gen}(msk)$. Let q be the number of oracle queries and let for $i \in \{1, \ldots, q\}$, (ek_i, dk_i) be the i-th answer from $\mathcal{O}_G$. We define the *(unrestricted strong) robustness advantage* of $\mathcal{A}$ as

$$\mathsf{Adv}^{\mathsf{sPKE\text{-}USROB}}_{\mathcal{E},\mathcal{A}} := \Pr\big[1 \le i_0, i_1 \le q \ \wedge \ i_0 \neq i_1 \\ \wedge \ \mathsf{Dec}\big(dk_{i_0}, \mathsf{San}(sp, c)\big) \neq \bot \neq \mathsf{Dec}\big(dk_{i_1}, \mathsf{San}(sp, c)\big)\big],$$

where the probability is over the randomness in $\mathsf{Exp}^{\mathsf{sPKE\text{-}USROB}}_{\mathcal{E},\mathcal{A}}$ and the random coins of San and Dec (both executed independently twice). The scheme $\mathcal{E}$ is *(unrestricted strongly) robust (USROB secure)* if $\mathsf{Adv}^{\mathsf{sPKE\text{-}USROB}}_{\mathcal{E},\mathcal{A}}$ is negligible for all efficient $\mathcal{A}$.

We next define IND-CCA security analogously to the definition for ordinary public-key encryption. In contrast to the usual definition, we do not require the adversary to output two messages of equal length, which implies that schemes satisfying our definition do not leak the length of the encrypted message.

Definition 5.3. Let $\mathcal{E} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{San}, \mathsf{Dec})$ be an sPKE scheme and let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a pair of probabilistic algorithms. Consider the experiment $\mathsf{Exp}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathcal{E},\mathcal{A}}$ in Figure 3 and let $C_{\mathcal{A}_2}$ be the set of all ciphertexts that $\mathcal{A}_2$ queried to the oracle $\mathcal{O}_{SD}$. We define the *ciphertext indistinguishability under chosen-ciphertext attacks advantage* of $\mathcal{A}$ as

$$\mathsf{Adv}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathcal{E},\mathcal{A}} := 2 \cdot \Pr\big[b' = b \ \wedge \ c^* \notin C_{\mathcal{A}_2}\big] - 1,$$

where the probability is over the randomness in $\mathsf{Exp}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathcal{E},\mathcal{A}}$. The scheme $\mathcal{E}$ has *indistinguishable ciphertexts under chosen-ciphertext attacks (is IND-CCA secure)* if $\mathsf{Adv}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathcal{E},\mathcal{A}}$ is negligible for all efficient $\mathcal{A}$.

We also need that it is hard to predict a ciphertext generated by Enc from a message of the adversary's choice given encryption and decryption keys. Note that this is not implied by IND-CCA security since the adversary here obtains the decryption key.

Definition 5.4. Let $\mathcal{E} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{San}, \mathsf{Dec})$ be an sPKE scheme and let $\mathcal{A}$ be a probabilistic algorithm. Consider the experiment $\mathsf{Exp}^{\mathsf{sPKE\text{-}UPD\text{-}CTXT}}_{\mathcal{E},\mathcal{A}}$ in Figure 3. We define the *ciphertext unpredictability advantage* of $\mathcal{A}$ as

$$\mathsf{Adv}^{\mathsf{sPKE\text{-}UPD\text{-}CTXT}}_{\mathcal{E},\mathcal{A}} := \Pr\big[c = c^*\big],$$

where the probability is over the randomness in $\mathsf{Exp}^{\mathsf{sPKE\text{-}UPD\text{-}CTXT}}_{\mathcal{E},\mathcal{A}}$. The scheme $\mathcal{E}$ has *unpredictable ciphertexts (is UPD-CTXT secure)* if $\mathsf{Adv}^{\mathsf{sPKE\text{-}UPD\text{-}CTXT}}_{\mathcal{E},\mathcal{A}}$ is negligible for all efficient $\mathcal{A}$.

We further define anonymity or indistinguishability of keys following Bellare et al. [BBDP01].

Definition 5.5. Let $\mathcal{E} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{San}, \mathsf{Dec})$ be an sPKE scheme and let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a pair of probabilistic algorithms. Consider the experiment $\mathsf{Exp}^{\mathsf{sPKE\text{-}IK\text{-}CCA}}_{\mathcal{E},\mathcal{A}}$ in Figure 3 and let $C_{\mathcal{A}_2}$ be the set of all ciphertexts that $\mathcal{A}_2$ queried to the oracle $\mathcal{O}_{SD_0}$ or $\mathcal{O}_{SD_1}$. We define the *indistinguishability of keys under chosen-ciphertext attacks advantage* of $\mathcal{A}$ as

$$\mathsf{Adv}^{\mathsf{sPKE\text{-}IK\text{-}CCA}}_{\mathcal{E},\mathcal{A}} := 2 \cdot \Pr\big[b' = b \ \wedge \ c^* \notin C_{\mathcal{A}_2}\big] - 1,$$

where the probability is over the randomness in $\mathsf{Exp}^{\mathsf{sPKE\text{-}IK\text{-}CCA}}_{\mathcal{E},\mathcal{A}}$. The scheme $\mathcal{E}$ has *indistinguishable keys under chosen-ciphertext attacks (is IK-CCA secure)* if $\mathsf{Adv}^{\mathsf{sPKE\text{-}IK\text{-}CCA}}_{\mathcal{E},\mathcal{A}}$ is negligible for all efficient $\mathcal{A}$.

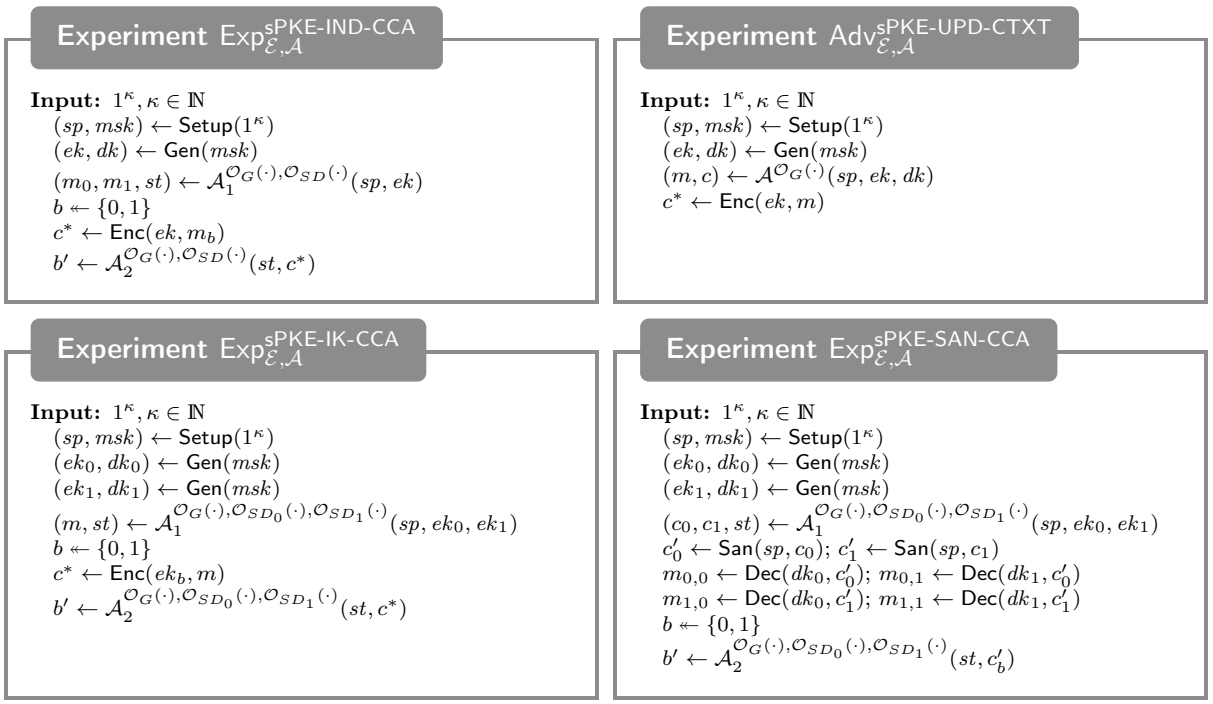

Figure 3: Security experiments for an sPKE scheme $\mathcal{E}$ and an adversary $\mathcal{A}$, where $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ in the experiments $\mathsf{Exp}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathcal{E},\mathcal{A}}$, $\mathsf{Exp}^{\mathsf{sPKE\text{-}IK\text{-}CCA}}_{\mathcal{E},\mathcal{A}}$, and $\mathsf{Exp}^{\mathsf{sPKE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}}$. The oracle $\mathcal{O}_{SD}$ is defined as $\mathcal{O}_{SD}(c) = \mathsf{Dec}(dk, \mathsf{San}(sp, c))$ and the oracle $\mathcal{O}_{SD_j}$ as $\mathcal{O}_{SD_j}(c) = \mathsf{Dec}(dk_j, \mathsf{San}(sp, c))$. Moreover, the oracle $\mathcal{O}_G$ on input `getNew`, outputs a fresh key pair $(ek, dk) \leftarrow \mathsf{Gen}(msk)$.

Sanitization security formalizes that given certain public keys and a sanitized ciphertext, it is hard to tell which of two adversarially chosen ciphertexts was actually sanitized. To exclude trivial attacks, we require that both ciphertexts are encryptions relative to the two challenge public keys ek_0 and ek_1. Otherwise, the adversary could use the oracle $\mathcal{O}_G$ to obtain a fresh key-pair (ek, dk) and encrypt two different messages under ek. It could then decrypt the challenge ciphertext using dk and win the game.

Definition 5.6. Let $\mathcal{E} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{San}, \mathsf{Dec})$ be an sPKE scheme and let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a pair of probabilistic algorithms. Consider the experiment $\mathsf{Exp}^{\mathsf{sPKE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}}$ in Figure 3. We define the *sanitization under chosen-ciphertext attacks advantage* of $\mathcal{A}$ as

$$\mathsf{Adv}^{\mathsf{sPKE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}} := 2 \cdot \Pr\left[b' = b \ \wedge \ \exists j, j' \in \{0,1\}\ m_{0,j} \neq \bot \neq m_{1,j'}\right] - 1,$$

where the probability is over the randomness in $\mathsf{Exp}^{\mathsf{sPKE\text{-}IK\text{-}CCA}}_{\mathcal{E},\mathcal{A}}$. The scheme $\mathcal{E}$ is *sanitization under chosen-ciphertext attacks (SAN-CCA) secure* if $\mathsf{Adv}^{\mathsf{sPKE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}}$ is negligible for all efficient $\mathcal{A}$.

We finally define the probability that two independent executions of the key-generation algorithm produce the same encryption key. This probability has to be small for all IND-CCA-secure schemes because an attacker can otherwise obtain a new key pair from $\mathcal{O}_G$ and if the obtained encryption key matches the one with which the challenge ciphertext is generated, the attacker can decrypt and win the IND-CCA game. We anyway explicitly define this probability to simplify our reductions later.

Definition 5.7. Let $\mathcal{E} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{San}, \mathsf{Dec})$ be an sPKE scheme. We define the *encryption-key collision probability* $\mathsf{Col}^{\mathsf{ek}}_{\mathcal{E}}$ as

$$\mathsf{Col}^{\mathsf{ek}}_{\mathcal{E}} := \Pr^{(sp,msk)\leftarrow\mathsf{Setup}(1^\kappa);\ (ek_0,dk_0)\leftarrow\mathsf{Gen}(msk);\ (ek_1,dk_1)\leftarrow\mathsf{Gen}(msk)}[ek_0 = ek_1].$$

5.2 Constructing an sPKE Scheme

We next construct an sPKE scheme satisfying our security definitions. Our construction resembles the weakly sanitizable PKE scheme by Fuchsbauer et al. [FGKO17]. We use a variant of ElGamal encryption and obtain security against chosen-ciphertext attacks using the technique of Naor and Yung [NY90], i.e., encrypting the message under two independent keys and proving in zero-knowledge that the ciphertexts are encryptions of the same message, which was shown to achieve full IND-CCA security if the zero-knowledge proof is one-time simulation sound by Sahai [Sah99].

Let PKE be a (IND-CPA secure) public-key encryption scheme, let Sig be a (EUF-CMA-secure) signature scheme, and let NIZK be a (one-time simulation sound) NIZK proof system for the language $L := \{x \mid \exists w\ (x,w) \in R\}$, where the relation R is defined as follows: for $x = \big(g, ek^{\mathsf{PKE}}, vk^{\mathsf{Sig}}, c_1, c_2, c_\sigma\big)$ and $w = (m, g^a, g^b, r_1, s_1, r_2, s_2, \sigma, r)$, we have $(x,w) \in R$ if and only if

$$c_1 = (g^{r_1}, g^{a\cdot r_1}, g^{s_1}, g^{a\cdot s_1}\cdot m) \ \wedge\ c_2 = (g^{r_2}, g^{b\cdot r_2}, g^{s_2}, g^{b\cdot s_2}\cdot m)$$
$$\wedge\ \mathsf{Sig.Ver}\big(vk^{\mathsf{Sig}}, (g^a, g^b), \sigma\big) = 1 \ \wedge\ c_\sigma = \mathsf{PKE.Enc}\big(ek^{\mathsf{PKE}}, (g^a, g^b, \sigma); r\big).$$

We define an sPKE scheme as follows:

Setup: The setup algorithm $\mathsf{sPKE.Setup}$ first generates

$$\big(ek^{\mathsf{PKE}}, dk^{\mathsf{PKE}}\big) \leftarrow \mathsf{PKE.Gen}(1^\kappa),$$
$$\big(vk^{\mathsf{Sig}}, sk^{\mathsf{Sig}}\big) \leftarrow \mathsf{Sig.Gen}(1^\kappa),$$
$$crs \leftarrow \mathsf{NIZK.Gen}(1^\kappa).$$

Let $G = \langle g\rangle$ be a cyclic group with prime order p generated by g, with $p \geq 2^\kappa$, and let $\mathcal{M} \subseteq G$ such that $|\mathcal{M}|/p \leq 2^{-\kappa}$. The sanitizer parameters sp^{sPKE} contain ek^{PKE}, vk^{Sig}, crs, and a description of G, including g and p. The master secret key msk^{sPKE} consists of ek^{PKE}, vk^{Sig}, sk^{Sig}, crs, and a description of G, including g and p.

Key generation: The algorithm $\mathsf{sPKE.Gen}$ on input msk^{sPKE}, samples two elements $dk_1, dk_2 \leftarrow\!\!\!\!\!\!\leftarrow \mathbb{Z}_p^*$ and computes $ek_1 \leftarrow g^{dk_1}$, $ek_2 \leftarrow g^{dk_2}$, as well as $\sigma \leftarrow \mathsf{Sig.Sign}\big(sk^{\mathsf{Sig}}, (ek_1, ek_2)\big)$. Finally, it outputs $ek^{\mathsf{sPKE}} := \big(g, p, crs, ek^{\mathsf{PKE}}, vk^{\mathsf{Sig}}, ek_1, ek_2, \sigma\big)$ and $dk^{\mathsf{sPKE}} := (dk_1, dk_2)$.

Encryption: The algorithm $\mathsf{sPKE.Enc}$ on input an encryption key $ek^{\mathsf{sPKE}} = \big(g, p, crs, ek^{\mathsf{PKE}}, vk^{\mathsf{Sig}}, ek_1, ek_2, \sigma\big)$ and a message $m \in \mathcal{M}$, samples randomness r, chooses $r_1, s_1, r_2, s_2 \leftarrow\!\!\!\!\!\!\leftarrow \mathbb{Z}_p^*$ uniformly at random, and computes

$$c_1 \leftarrow \big(g^{r_1}, ek_1^{r_1}, g^{s_1}, ek_1^{s_1}\cdot m\big),$$
$$c_2 \leftarrow \big(g^{r_2}, ek_2^{r_2}, g^{s_2}, ek_2^{s_2}\cdot m\big),$$
$$c_\sigma \leftarrow \mathsf{PKE.Enc}\big(ek^{\mathsf{PKE}}, (ek_1, ek_2, \sigma); r\big).$$

It then generates $\pi \leftarrow \mathsf{NIZK.Prove}\big(crs, x := (g, ek^{\mathsf{PKE}}, vk^{\mathsf{Sig}}, c_1, c_2, c_\sigma), w := (m, ek_1, ek_2, r_1, s_1, r_2, s_2, \sigma, r)\big)$. It finally outputs the ciphertext $c := (c_1, c_2, c_\sigma, \pi)$.

Sanitization: The algorithm $\mathsf{sPKE.San}$ on input sanitizer parameters sp^{sPKE} and a ciphertext $c = (c_1, c_2, c_\sigma, \pi)$, first verifies the NIZK proof by evaluating $\mathsf{NIZK.Ver}\big(crs, x \coloneqq (g, ek^{\mathsf{PKE}}, vk^{\mathsf{Sig}}, c_1, c_2, c_\sigma), \pi\big)$. It then parses $(c_{1,1}, c_{1,2}, c_{1,3}, c_{1,4}) \leftarrow c_1$. If the verification succeeds and $c_{1,1} \neq 1 \neq c_{1,2}$, then it chooses a random $t \twoheadleftarrow \mathbb{Z}_p^*$ and outputs the sanitized ciphertext

$$c' \coloneqq \big((c_{1,1})^t \cdot c_{1,3},\ (c_{1,2})^t \cdot c_{1,4}\big).$$

If the verification fails or if $c_{1,1} = 1$ or $c_{1,2} = 1$, it outputs $\perp$.

Decryption: The algorithm $\mathsf{sPKE.Dec}$ on input a decryption key $dk^{\mathsf{sPKE}} = (dk_1, dk_2)$ and a sanitized ciphertext $c' = (c'_1, c'_2)$, computes the message $m \leftarrow c'_2 \cdot \big((c'_1)^{dk_1}\big)^{-1}$. It outputs m if $m \in \mathcal{M}$, and otherwise it outputs $\perp$. On input dk^{sPKE} and $\perp$, it outputs $\perp$.

We first prove correctness and other straightforward properties of the scheme.

Proposition 5.8. *If* Sig *is correct and* NIZK *has perfect completeness, the scheme* sPKE *from above is correct, robust, has unpredictable ciphertexts, and negligible encryption-key collision probability.*

Proof. To prove correctness, let $(sp^{\mathsf{sPKE}}, msk^{\mathsf{sPKE}})$ in the range of $\mathsf{sPKE.Setup}$, $(ek^{\mathsf{sPKE}}, dk^{\mathsf{sPKE}})$ in the range of $\mathsf{sPKE.Gen}(msk^{\mathsf{sPKE}})$, and let $m \in \mathcal{M}$. By correctness of Sig and completeness of NIZK, the NIZK verification in $\mathsf{sPKE.San}$ in the correctness experiment succeeds with probability 1. Moreover, since g generates G and $r_1, dk_1 \in \mathbb{Z}_p^*$, we have $c_{1,1} = g^{r_1} \neq 1$ and $c_{1,2} = ek_1^{r_1} = g^{dk_1 \cdot r_1} \neq 1$. Hence,

$$\begin{aligned} c' &= \mathsf{sPKE.San}\big(sp^{\mathsf{sPKE}}, \mathsf{sPKE.Enc}(ek^{\mathsf{sPKE}}, m)\big) = \big((c_{1,1})^t \cdot c_{1,3},\ (c_{1,2})^t \cdot c_{1,4}\big) \\ &= \big(g^{r_1 \cdot t + s_1},\ ek_1^{r_1 \cdot t + s_1} \cdot m\big). \end{aligned}$$

and

$$\mathsf{sPKE.Dec}\big(dk^{\mathsf{sPKE}}, c'\big) = ek_1^{r_1 \cdot t + s_1} \cdot m \cdot \Big(\big(g^{r_1 \cdot t + s_1}\big)^{dk_1}\Big)^{-1} = g^{dk_1(r_1 \cdot t + s_1)} \cdot m \cdot \Big(g^{dk_1(r_1 \cdot t + s_1)}\Big)^{-1} = m.$$

This shows that sPKE is correct.

For ciphertext unpredictability, note that each ciphertext contains g^{r_1}, g^{s_1}, g^{r_2}, and g^{s_2} for uniformly chosen $r_1, s_1, r_2, s_2 \in \mathbb{Z}_p^*$. Each of these elements can only be guessed with probability $1/|\mathbb{Z}_p^*| = 1/(p-1)$, where $p \geq 2^\kappa$. We can therefore conclude that for any $\mathcal{A}$,

$$\mathsf{Adv}^{\mathsf{sPKE\text{-}UPD\text{-}CTXT}}_{\mathsf{sPKE},\mathcal{A}} \leq \frac{1}{(p-1)^4} \leq \frac{1}{(2^\kappa - 1)^4}.$$

Similarly, since the encryption keys contain the pairs $(ek_1 = g^{dk_1}, ek_2 = g^{dk_2})$ for uniformly chosen $dk_1, dk_2 \in \mathbb{Z}_p^*$, we have

$$\mathsf{Col}^{\mathsf{ek}}_{\mathsf{sPKE}} \leq \frac{1}{(p-1)^2} \leq \frac{1}{(2^\kappa - 1)^2}.$$

We finally prove robustness. To this end, let $\mathcal{A}$ be a probabilistic algorithm that makes at most q queries to $\mathcal{O}_G$ and consider $\mathsf{Exp}^{\mathsf{sPKE\text{-}USROB}}_{\mathcal{E},\mathcal{A}}$. Further let ek_i^{sPKE} and $dk_i^{\mathsf{sPKE}} = (dk_{i,1}, dk_{i,2})$ be the keys returned from $\mathcal{O}_G$ for the i-th query and let (c, i_0, i_1) be the output of $\mathcal{A}$, where $c \coloneqq (c_1, c_2, c_\sigma, \pi)$ and $c_1 = (g^a, g^b, g^c, g^d)$. Assume that $i_0 \neq i_1$ and that c passes sanitization,

since $\mathcal{A}$ cannot win otherwise. This implies $a \neq 0 \neq b$ and sanitizing and decrypting the ciphertext with the two decryption keys yield $m_0 = g^{bt_0+d-dk_{i_0,1}(at_0+c)}$ and $m_1 = g^{bt_1+d-dk_{i_1,1}(at_1+c)}$, respectively, where $t_0, t_1 \in \mathbb{Z}_p^*$ are chosen uniformly during sanitization. We then have that $\mathcal{A}$ wins if $m_0, m_1 \in \mathcal{M}$. Assume that $m_0 \in \mathcal{M}$. Then,

$$\begin{aligned} m_1 &= m_0 \cdot g^{-b\cdot t_0-d+dk_{i_0,1}(a\cdot t_0+c)} \cdot g^{b\cdot t_1+d-dk_{i_1,1}(a\cdot t_1+c)} \\ &= m_0 \cdot g^{(dk_{i_0,1}-dk_{i_1,1})c} \cdot g^{(a\cdot dk_{i_0,1}-b)t_0} \cdot g^{(b-a\cdot dk_{i_1,1})t_1}. \end{aligned}$$

Note that if $dk_{i_0,1} \neq dk_{i_1,1}$ and $a \neq 0 \neq b$, then $a\cdot dk_{i_0,1}-b$ and $b-a\cdot dk_{i_1,1}$ cannot both be 0. Hence, in this case, m_1 is a uniformly random element in the group G. The probability that $m_1 \in \mathcal{M}$ is therefore $|\mathcal{M}|/|G| \leq 2^{-\kappa}$. Since $\mathcal{A}$ obtains at most q decryption keys and the $dk_{i,1}$ are uniform elements in $\mathbb{Z}_p^*$, the probability that $dk_{i_0,1} = dk_{i_1,1}$ is bounded by $q^2 \cdot 1/(p-1) \leq q^2 \cdot 1/(2^\kappa - 1)$. We can therefore conclude that

$$\mathsf{Adv}^{\mathsf{sPKE\text{-}USROB}}_{\mathsf{sPKE},\mathcal{A}} \leq 2^{-\kappa} + \frac{q^2}{2^\kappa - 1} \leq \frac{q^2+1}{2^\kappa - 1}. \qquad \square$$

The main result of this section is the security of the scheme, summarized in the following theorem.

Theorem 5.9. *If the DDH assumption holds in the group G, PKE is IND-CPA secure, Sig is EUF-CMA secure, and if NIZK is zero-knowledge, computationally sound, and one-time simulation sound, then the scheme sPKE from above is IND-CCA secure, IK-CCA secure, and SAN-CCA secure.*

On a high level, our proof proceeds as follows. It is rather straightforward to show that our variant of ElGamal encryption satisfies the CPA versions of the three properties. The proof of CCA security follows the proof by Sahai for public-key encryption [Sah99]: Since the NIZK ensures that both ciphertext components are encryptions of the same message, it does not matter which component is decrypted. In a reduction, where we assume an adversary $\mathcal{A}$ against the CCA variants of the desired properties, and we want to break the corresponding CPA variants, we only get one public key and no decryption oracle from the challenger. In order to emulate the view toward $\mathcal{A}$, the reduction chooses an additional public key and a CRS for the NIZK scheme. Since the reduction thus knows one of the secret keys, it can emulate a decryption oracle. To generate a challenge ciphertext, the reduction obtains one challenge ciphertexts from its CPA challenger, and encrypts another, arbitrary message to get a second ciphertext. The reduction uses the NIZK simulator to obtain an accepting proof that is indistinguishable from a "real proof", even if the underlying statement is not true. A crucial point here is that the NIZK scheme has to be *one-time simulation sound* (see Definition A.11). This ensures that even if the adversary sees one simulated (accepting) proof of a wrong statement, it is not capable of producing accepting proofs of wrong statements, except by reproducing the exact proof obtained within the challenge, but which $\mathcal{A}$ is not allowed to ask to the decryption oracle by the CCA definition. The fundamental result of Sahai [Sah99] is that the above strategy successfully simulates a complete CCA attack toward $\mathcal{A}$.

An additional obstacle we have is that to preserve anonymity, the NIZK needs to be verified without knowing which encryption keys were used. On the other hand, the reduction only works if the two used keys "match", since otherwise, the emulated decryption oracle would use an incorrect key to decrypt. To prevent an adversary from mixing different key pairs for encryptions, the key-generation process signs valid key pairs, and the NIZK ensures that a signed pair was

used. Due to anonymity, this signature cannot be directly contained in the ciphertexts. Instead, it is part of the witness. To prove that if a ciphertext is accepted, the used key pair was indeed signed by the key-generation process, we show that if $\mathcal{A}$ manages to produce a ciphertext that is accepted but the keys were not signed, we can break EUF-CMA security of the signature scheme. In this reduction, we have to provide a forgery. Hence, the reduction needs to extract the signature and the used encryption keys from the ciphertext. This could be achieved by assuming that the NIZK is extractable. Extractability and simulation-soundness at the same time is, however, a quite strong assumption. Instead, we add an encryption of the signature and the key pair under a separate PKE scheme to the ciphertexts. The reduction can then generate the keys for this PKE scheme itself and perform extraction by decrypting that ciphertext.

Since the proofs for IND-CCA and IK-CCA security closely follow the proof by Sahai [Sah99], we here prove SAN-CCA security and defer the other proofs to Appendix C.

Lemma 5.10. *Let* sPKE *be the scheme from above and let* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *be a pair of probabilistic algorithms such that* $\mathcal{A}_1$ *and* $\mathcal{A}_2$ *together make at most* q_G *queries to* $\mathcal{O}_G$ *and at most* q_{SD} *queries to* $\mathcal{O}_{SD_0}$ *and* $\mathcal{O}_{SD_1}$ *combined. Then, there exist adversaries* $\mathcal{A}_{\mathsf{DDH}}$, $\mathcal{A}_{\mathsf{snd}}$, *and* $\mathcal{A}_{\mathsf{Sig}}$ *(which are all roughly as efficient as emulating an execution of* $\mathsf{Exp}^{\mathsf{sPKE\text{-}SAN\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}}$*) such that*

$$\mathsf{Adv}^{\mathsf{sPKE\text{-}SAN\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}} \leq 8 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{g,\mathcal{A}_{\mathsf{DDH}}} + (24q_{SD} + 48) \cdot \mathsf{Adv}^{\mathsf{NIZK\text{-}snd}}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{snd}}} + 24 \cdot \mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}}} + \frac{52q_G^2 + 192q_G + 196}{2^\kappa - 1}.$$

Proof. Let W_{san} be the event that $\mathcal{A}$ wins the sanitization game, i.e.,

$$W_{\text{san}} := \left[b' = b \ \wedge \ \exists j, j' \in \{0,1\} \ m_{0,j} \neq \bot \neq m_{1,j'}\right].$$

We define hybrid experiments H_0 to H_2 as follows:

- $H_0 := \mathsf{Exp}^{\mathsf{sPKE\text{-}SAN\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}}$ is the sanitization experiment.
- H_1 is identical to H_0, except that if $c_0' \neq \bot$, then c_0' is replaced by two uniformly random group elements (g^b, g^c).
- H_2 is identical to H_1, except that if $c_1' \neq \bot$, then c_1' is replaced by two uniformly random group elements (g^b, g^c).

In H_2, if $c_0' \neq \bot$ and $c_1' \neq \bot$, the view of $\mathcal{A}$ is independent of the bit b. Hence, $\mathcal{A}$ cannot guess b with probability more than $1/2$ in this case. On the other hand, if $c_0' = \bot$ or $c_1' = \bot$, then $m_{0,0} = m_{0,1} = \bot$ or $m_{1,0} = m_{1,1} = \bot$, respectively, since $\bot$ decrypts to $\bot$. By definition of the sanitization advantage, $\mathcal{A}$ cannot win in this case. Thus,

$$\Pr^{H_2}[W_{\text{san}}] \leq \frac{1}{2}. \tag{2}$$

To conclude the proof, we show that the probability of W_{san} in H_0 differs only negligibly from its probability in H_2. To this end, we first prove that three bad events occur only with negligible probability in any of the hybrids.

Claim 1. *Let* $i \in \{0,1,2\}$ *and consider the experiment* H_i*. Further let* B_1 *be the event that* $\mathcal{A}$ *outputs as* c_0 *or* c_1 *or queries at least one of its decryption oracles with a valid but improper*

ciphertext $(c_1, c_2, c_\sigma, \pi)$*, i.e.,* $\big(g, ek^{\mathsf{PKE}}, vk^{\mathsf{Sig}}, c_1, c_2, c_\sigma\big) \notin L$*, but where* π *is an accepting proof, i.e.,* $\mathsf{NIZK.Ver}\big(crs, x \coloneqq (g, ek^{\mathsf{PKE}}, vk^{\mathsf{Sig}}, c_1, c_2, c_\sigma), \pi\big) = 1$*. Then, there exists an adversary* $\mathcal{A}^i_{\mathsf{snd}}$ *such that*

$$\Pr^{H_i}[B_1] \leq (q_{SD} + 2) \cdot \mathsf{Adv}^{\mathsf{NIZK\text{-}snd}}_{\mathsf{NIZK},\mathcal{A}^i_{\mathsf{snd}}}.$$

Proof of claim. On input *crs* from the soundness challenger, $\mathcal{A}^i_{\mathsf{snd}}$ uses this CRS, generates all needed keys itself, and emulates an execution of H_i toward $\mathcal{A}$. It initially chooses $q_0 \twoheadleftarrow \{-1, 0, 1, \ldots, q_{SD}\}$ uniformly at random. If $q_0 > 0$ and when $\mathcal{A}$ submits the q_0-th query to a decryption oracle, $\mathcal{A}^i_{\mathsf{snd}}$ outputs the corresponding statement and proof to the challenger. If $q_0 \leq 0$ and when $\mathcal{A}$ outputs (c_0, c_1, st), then $\mathcal{A}^i_{\mathsf{snd}}$ submits the statement and proof from c_{q_0+1} to the challenger. If B_1 occurs, then for some q_0, $\mathcal{A}^i_{\mathsf{snd}}$ outputs an accepting proof for an incorrect statement. Hence, the claim follows. ◊

Claim 2. *Let* $i \in \{0, 1, 2\}$ *and consider the experiment* H_i*. Further let* B_2 *be the event that* $\mathcal{A}$ *outputs as* c_0 *or* c_1 *or queries at least one of its decryption oracles with a valid and proper ciphertext* $(c_1, c_2, c_\sigma, \pi)$*, i.e.,* $\big(g, ek^{\mathsf{PKE}}, vk^{\mathsf{Sig}}, c_1, c_2, c_\sigma\big) \in L$ *and* π *is accepting, but where* c_σ *is the encryption of a triple* (ek_1, ek_2, σ)*, such that the pair* (ek_1, ek_2) *has never been output by the experiment or the oracle* $\mathcal{O}_G$*. Then, there exists an adversary* $\mathcal{A}^i_{\mathsf{Sig}}$ *such that*

$$\Pr^{H_i}[B_2] \leq \mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}^i_{\mathsf{Sig}}}.$$

Proof of claim. On input a signature verification key vk^{Sig}, $\mathcal{A}^i_{\mathsf{Sig}}$ generates all keys except for vk^{Sig} and sk^{Sig}, and emulates an execution of H_i. To generate the encryption keys ek_0^{sPKE} and ek_1^{sPKE} and to answer queries to $\mathcal{O}_G$, $\mathcal{A}^i_{\mathsf{Sig}}$ obtains the needed signature using the signing oracle of $\mathsf{Exp}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}^i_{\mathsf{Sig}}}$. The rest of H_i is straightforward to emulate since $\mathcal{A}^i_{\mathsf{Sig}}$ possesses all keys except for sk^{Sig}. Whenever $\mathcal{A}$ returns or submits a ciphertext $(c_1, c_2, c_\sigma, \pi)$ to one of the decryption oracles, $\mathcal{A}^i_{\mathsf{Sig}}$ decrypts c_σ to obtain a pair (ek'_1, ek'_2) and a signature σ'. If it has never queried (ek'_1, ek'_2) to its signing oracle and if the signature is valid, then it outputs $((ek'_1, ek'_2), \sigma')$ as its forgery. Note that if B_2 occurs, $\mathcal{A}^i_{\mathsf{Sig}}$ obtains a forgery, so the claim follows. ◊

Claim 3. *Let* $i \in \{0, 1, 2\}$ *and consider the experiment* H_i*. Further let* B_3 *be the event that* H_i *generates two different encryption keys* $ek^{\mathsf{sPKE}} = \big(g, p, crs, ek^{\mathsf{PKE}}, vk^{\mathsf{Sig}}, ek_1, ek_2, \sigma\big)$ *and* $\big(ek^{\mathsf{sPKE}}\big)' = \big(g, p, crs, ek^{\mathsf{PKE}}, vk^{\mathsf{Sig}}, ek'_1, ek'_2, \sigma'\big)$ *such that* $ek_1 = ek'_1$ *or* $ek_2 = ek'_2$*. Then,*

$$\Pr^{H_i}[B_3] \leq \frac{2(q_G + 2)^2}{2^\kappa - 1}.$$

Proof of claim. The experiment H_i initially generates two encryption keys and then one for each query to $\mathcal{O}_G$. Hence, there are at most $(q_G + 2)^2$ such pairs. For each of these pairs, the probability that one of the two components collides is at most $2 \cdot (1/|\mathbb{Z}^*_p|) = 2/(p-1)$. Using $p \geq 2^\kappa$ and the union bound implies the claim. ◊

We now bound the difference of the probabilities of W_{san} in different hybrids. To this end, let $B \coloneqq B_1 \cup B_2 \cup B_3$.

Claim 4. *For all* $i \in \{0, 1\}$*, there exists an adversary* $\mathcal{A}^i_{\mathsf{DDH}}$ *such that*

$$\Pr^{H_i}[W_{\mathsf{san}}] - \Pr^{H_{i+1}}[W_{\mathsf{san}}] \leq 2 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{g,\mathcal{A}^i_{\mathsf{DDH}}} + 2 \cdot \Pr^{H_i}[B] + 4 \cdot \Pr^{H_{i+1}}[B] + \frac{q_G^2 + 1}{2^\kappa - 1}.$$

Proof of claim. Let $i \in \{0,1\}$ and let G_0 and G_1 be the events that c_i output by $\mathcal{A}$ is an encryption under ek_0^{sPKE} and ek_1^{sPKE}, respectively. If B, G_0, and G_1 all do not occur, then c_i is either invalid or a valid encryption under a key different from ek_0^{sPKE} and ek_1^{sPKE}. Since W_{san} can only occur if the ciphertext decrypts to a message different from $\perp$ under one of these keys, this only happens if robustness is violated. Using the result on robustness derived in the proof of Proposition 5.8, this implies

$$\Pr^{H_i}[W_{\mathsf{san}} \cap \neg B \cap \neg G_1 \cap \neg G_2] \leq \Pr^{H_i}[W_{\mathsf{san}} \mid \neg B \cap \neg G_1 \cap \neg G_2] \leq \frac{q_G^2 + 1}{2^\kappa - 1}.$$

We also have

$$\begin{aligned}
&\Pr^{H_i}[W_{\mathsf{san}}] - \Pr^{H_{i+1}}[W_{\mathsf{san}}] \\
&= \Pr^{H_i}[W_{\mathsf{san}} \cap \neg B \cap (G_1 \cup G_2)] + \Pr^{H_i}[W_{\mathsf{san}} \cap (B \cup \neg(G_1 \cup G_2))] \\
&\quad - \Pr^{H_{i+1}}[W_{\mathsf{san}} \cap \neg B \cap (G_1 \cup G_2)] - \Pr^{H_{i+1}}[W_{\mathsf{san}} \cap (B \cup \neg(G_1 \cup G_2))] \\
&\leq \Pr^{H_i}[W_{\mathsf{san}} \cap \neg B \cap (G_1 \cup G_2)] - \Pr^{H_{i+1}}[W_{\mathsf{san}} \cap \neg B \cap (G_1 \cup G_2)] \\
&\quad + \Pr^{H_i}[B] + \Pr^{H_i}[W_{\mathsf{san}} \cap \neg(G_1 \cup G_2)],
\end{aligned}$$

and

$$\Pr^{H_i}[W_{\mathsf{san}} \cap \neg(G_1 \cup G_2)] \leq \Pr^{H_i}[W_{\mathsf{san}} \cap \neg B \cap \neg(G_1 \cup G_2)] + \Pr^{H_i}[B].$$

This implies

$$\begin{aligned}
\Pr^{H_i}[W_{\mathsf{san}}] - \Pr^{H_{i+1}}[W_{\mathsf{san}}] \leq \Pr^{H_i}[W_{\mathsf{san}} \cap \neg B \cap (G_1 \cup G_2)] \\
- \Pr^{H_{i+1}}[W_{\mathsf{san}} \cap \neg B \cap (G_1 \cup G_2)] + 2 \cdot \Pr^{H_i}[B] + \frac{q_G^2 + 1}{2^\kappa - 1}. \quad (3)
\end{aligned}$$

We now define the adversary $\mathcal{A}^i_{\mathsf{DDH}}$. On input (X, Y, T), $\mathcal{A}^i_{\mathsf{DDH}}$ chooses $j \twoheadleftarrow \{0,1\}$ uniformly at random and sets $ek_{j,1} \leftarrow X$. All remaining keys, including $ek_{j,2}$, are generated as in H_i, and $\mathcal{A}$ is invoked on $\big(sp^{\mathsf{sPKE}}, ek_0^{\mathsf{sPKE}} = (ek_{0,1}, ek_{0,2}), ek_1^{\mathsf{sPKE}} = (ek_{1,1}, ek_{1,2})\big)$. The adversary $\mathcal{A}^i_{\mathsf{DDH}}$ then emulates an execution of H_i. Since it has all keys except for the decryption key $dk_{j,1}$, only the emulation of the decryption oracle $\mathcal{O}_{SD_j}$ is nontrivial. To answer queries to this oracle, $\mathcal{A}^i_{\mathsf{DDH}}$ sanitizes and decrypts the second ciphertext component instead of the first one using $dk_{j,2}$. When $\mathcal{A}$ outputs $\big(c_0, c_1, st\big)$, both ciphertexts are sanitized and decrypted as in the emulation of the decryption oracles, except that $m_{i,j}$ is not set to $\perp$ during decryption if $m_{i,j} \notin \mathcal{M}$. If $c_i' \neq \perp$, it is replaced by $c_i' \leftarrow (Y, T \cdot m_{i,j})$. Moreover, $\mathcal{A}^i_{\mathsf{DDH}}$ decrypts $c_{i,\sigma}$ and checks whether it contains the encryption keys corresponding to ek_j^{sPKE}. If this is not the case, it terminates and returns 0. Otherwise, it continues with the emulation. Finally, when $\mathcal{A}$ terminates, $\mathcal{A}^i_{\mathsf{DDH}}$ outputs $d = 1$ if W_{san} occurs, and $d = 0$ otherwise.

Note that B not occurring implies that $\mathcal{A}^i_{\mathsf{DDH}}$ emulates the decryption oracle perfectly since in this case, all submitted valid ciphertexts contain two encryptions of the same message under a signed key pair. Moreover, due to $\neg B_3$, the first encryption key matches the first key of the oracle if and only if the second keys match. If they match, decryption with either key yields the correct message with probability 1. Otherwise, the message (before potentially being set to $\perp$) is a uniform group element for both keys, as shown in the robustness proof of Proposition 5.8.

Furthermore, if (X, Y, T) are three independent uniform group elements, c_i' gets replaced by two uniformly random group elements if $c_i' \neq \perp$, as in H_{i+1}. On the other hand, if $\neg B$ and

G_j occur and if $c'_i \neq \perp$, then $c_i = (c_{i,1}, c_{i,2}, c_{i,\sigma}, \pi_i)$ is a valid encryption of $m_{i,j}$ under ek_j^{sPKE}. Hence, there exist $r_1, s_1 \in \mathbb{Z}_p^*$ such that

$$c_{i,1} = \left(g^{r_1}, (ek_{j,1})^{r_1}, g^{s_1}, (ek_{j,1})^{s_1} \cdot m_{i,j}\right) = \left(g^{r_1}, X^{r_1}, g^{s_1}, X^{s_1} \cdot m_{i,j}\right).$$

In H_i, this ciphertext is sanitized to $c'_i = \left(g^{r_1 \cdot t + s_1}, X^{r_1 \cdot t + s_1} \cdot m_{i,j}\right)$ for $t \twoheadleftarrow \mathbb{Z}_p^*$. If we further have $X = g^a$, $Y = g^b$, and $T = g^{ab}$, then this corresponds to $c'_i = \left(g^{r_1 \cdot t + s_1}, g^{a \cdot (r_1 \cdot t + s_1)} \cdot m_{i,j}\right)$, which is equally distributed as the sanitization $(Y, T \cdot m_{i,j})$ generated by $\mathcal{A}^i_{\mathsf{DDH}}$. Since we also have that the probability of $\neg B \cap G_j$ is equal in $\mathsf{DDH}^{\mathrm{real}}_{g,\mathcal{A}^i_{\mathsf{DDH}}}$ and H_i, as well as in $\mathsf{DDH}^{\mathrm{rand}}_{g,\mathcal{A}^i_{\mathsf{DDH}}}$ and H_{i+1}, we can conclude

$$\Pr^{\mathsf{DDH}^{\mathrm{real}}_{g,\mathcal{A}^i_{\mathsf{DDH}}}}[d = 1 \cap \neg B \cap G_j] = \Pr^{H_i}[W_{\mathrm{san}} \cap \neg B \cap G_j],$$
$$\Pr^{\mathsf{DDH}^{\mathrm{rand}}_{g,\mathcal{A}^i_{\mathsf{DDH}}}}[d = 1 \cap \neg B \cap G_j] = \Pr^{H_{i+1}}[W_{\mathrm{san}} \cap \neg B \cap G_j].$$

Hence,

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{DDH}}_{g,\mathcal{A}^i_{\mathsf{DDH}}} &= \Pr^{\mathsf{DDH}^{\mathrm{real}}_{g,\mathcal{A}^i_{\mathsf{DDH}}}}[d = 1] - \Pr^{\mathsf{DDH}^{\mathrm{rand}}_{g,\mathcal{A}^i_{\mathsf{DDH}}}}[d = 1] \\
&= \Pr^{\mathsf{DDH}^{\mathrm{real}}_{g,\mathcal{A}^i_{\mathsf{DDH}}}}[d = 1 \cap \neg B \cap G_j] + \Pr^{\mathsf{DDH}^{\mathrm{real}}_{g,\mathcal{A}^i_{\mathsf{DDH}}}}[d = 1 \cap (B \cup \neg G_j)] \\
&\quad - \Pr^{\mathsf{DDH}^{\mathrm{rand}}_{g,\mathcal{A}^i_{\mathsf{DDH}}}}[d = 1 \cap \neg B \cap G_j] - \Pr^{\mathsf{DDH}^{\mathrm{rand}}_{g,\mathcal{A}^i_{\mathsf{DDH}}}}[d = 1 \cap (B \cup \neg G_j)] \\
&\geq \Pr^{H_i}[W_{\mathrm{san}} \cap \neg B \cap G_j] - \Pr^{H_{i+1}}[W_{\mathrm{san}} \cap \neg B \cap G_j] \\
&\quad - \Pr^{\mathsf{DDH}^{\mathrm{rand}}_{g,\mathcal{A}^i_{\mathsf{DDH}}}}[d = 1 \cap B] - \Pr^{\mathsf{DDH}^{\mathrm{rand}}_{g,\mathcal{A}^i_{\mathsf{DDH}}}}[d = 1 \cap \neg G_j].
\end{aligned}$$

If $\neg B$ occurs, then $c_{i,\sigma}$ contains the correct encryption keys and thus, if also $\neg G_j$ occurs, $\mathcal{A}^i_{\mathsf{DDH}}$ always returns 0. This implies $\Pr^{\mathsf{DDH}^{\mathrm{rand}}_{g,\mathcal{A}^i_{\mathsf{DDH}}}}[d = 1 \cap \neg G_j \cap \neg B] = 0$, and therefore

$$\begin{aligned}
\Pr^{\mathsf{DDH}^{\mathrm{rand}}_{g,\mathcal{A}^i_{\mathsf{DDH}}}}[d = 1 \cap \neg G_j] &= \Pr^{\mathsf{DDH}^{\mathrm{rand}}_{g,\mathcal{A}^i_{\mathsf{DDH}}}}[d = 1 \cap \neg G_j \cap \neg B] + \Pr^{\mathsf{DDH}^{\mathrm{rand}}_{g,\mathcal{A}^i_{\mathsf{DDH}}}}[d = 1 \cap \neg G_j \cap B] \\
&\leq \Pr^{\mathsf{DDH}^{\mathrm{rand}}_{g,\mathcal{A}^i_{\mathsf{DDH}}}}[B].
\end{aligned}$$

Using $\Pr^{\mathsf{DDH}^{\mathrm{rand}}_{g,\mathcal{A}^i_{\mathsf{DDH}}}}[B] = \Pr^{H_{i+1}}[B]$, we obtain

$$\mathsf{Adv}^{\mathsf{DDH}}_{g,\mathcal{A}^i_{\mathsf{DDH}}} \geq \Pr^{H_i}[W_{\mathrm{san}} \cap \neg B \cap G_j] - \Pr^{H_{i+1}}[W_{\mathrm{san}} \cap \neg B \cap G_j] - 2 \cdot \Pr^{H_{i+1}}[B].$$

Combining this with equation (3) and the fact that given $G_1 \cup G_2$ and $\neg B$, G_j occurs with probability $1/2$ (independently of W_{san}), we can conclude

$$\begin{aligned}
\Pr^{H_i}[W_{\mathsf{san}}] - \Pr^{H_{i+1}}[W_{\mathsf{san}}] &\leq 2 \cdot \Pr^{H_i}[W_{\mathsf{san}} \cap \neg B \cap G_j] - 2 \cdot \Pr^{H_{i+1}}[W_{\mathsf{san}} \cap \neg B \cap G_j] \\
&\quad + 2 \cdot \Pr^{H_i}[B] + \frac{q_G^2 + 1}{2^\kappa - 1} \\
&\leq 2 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{g,\mathcal{A}^i_{\mathsf{DDH}}} + 4 \cdot \Pr^{H_{i+1}}[B] + 2 \cdot \Pr^{H_i}[B] + \frac{q_G^2 + 1}{2^\kappa - 1}. \qquad \diamond
\end{aligned}$$

Using Claim 4 and equation (2), we obtain

$$\begin{aligned}
&\mathsf{Adv}^{\mathsf{sPKE\text{-}SAN\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}} \\
&= 2 \cdot \Pr^{H_0}[W_{\mathrm{san}}] - 1 \\
&= 2 \cdot \left(\Pr^{H_0}[W_{\mathrm{san}}] - \Pr^{H_1}[W_{\mathrm{san}}] + \Pr^{H_1}[W_{\mathrm{san}}] - \Pr^{H_2}[W_{\mathrm{san}}] + \Pr^{H_2}[W_{\mathrm{san}}]\right) - 1 \\
&\leq 4 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{g,\mathcal{A}^0_{\mathsf{DDH}}} + 4 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{g,\mathcal{A}^1_{\mathsf{DDH}}} + 4 \cdot \Pr^{H_0}[B] + 12 \cdot \Pr^{H_1}[B] + 8 \cdot \Pr^{H_2}[B] + 4 \cdot \frac{q_G^2 + 1}{2^\kappa - 1}.
\end{aligned}$$

Claims 1 to 3 further imply

$$\Pr^{H_i}[B] \leq (q_{SD} + 2) \cdot \mathsf{Adv}^{\mathsf{NIZK\text{-}snd}}_{\mathsf{NIZK},\mathcal{A}^i_{\mathsf{snd}}} + \mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}^i_{\mathsf{Sig}}} + \frac{2(q_G + 2)^2}{2^\kappa - 1}.$$

Hence,

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{sPKE\text{-}SAN\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}} \leq\ & 4 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{g,\mathcal{A}^0_{\mathsf{DDH}}} + 4 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{g,\mathcal{A}^1_{\mathsf{DDH}}} + (4q_{SD} + 8) \cdot \mathsf{Adv}^{\mathsf{NIZK\text{-}snd}}_{\mathsf{NIZK},\mathcal{A}^0_{\mathsf{snd}}} \\
&+ (12q_{SD} + 24) \cdot \mathsf{Adv}^{\mathsf{NIZK\text{-}snd}}_{\mathsf{NIZK},\mathcal{A}^1_{\mathsf{snd}}} + (8q_{SD} + 16) \cdot \mathsf{Adv}^{\mathsf{NIZK\text{-}snd}}_{\mathsf{NIZK},\mathcal{A}^2_{\mathsf{snd}}} \\
&+ 4 \cdot \mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}^0_{\mathsf{Sig}}} + 12 \cdot \mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}^1_{\mathsf{Sig}}} + 8 \cdot \mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}^2_{\mathsf{Sig}}} \\
&+ \frac{48(q_G + 2)^2 + 4q_G^2 + 4}{2^\kappa - 1}.
\end{aligned}$$

We define the adversary $\mathcal{A}_{\mathsf{DDH}}$ as running $\mathcal{A}^0_{\mathsf{DDH}}$ and $\mathcal{A}^1_{\mathsf{DDH}}$ with probability $\frac{1}{2}$ each, the adversary $\mathcal{A}_{\mathsf{snd}}$ as running $\mathcal{A}^0_{\mathsf{snd}}$ with probability $\frac{4q_{SD}+8}{24q_{SD}+48}$, $\mathcal{A}^1_{\mathsf{snd}}$ with probability $\frac{12q_{SD}+24}{24q_{SD}+48}$, and $\mathcal{A}^2_{\mathsf{snd}}$ with probability $\frac{8q_{SD}+16}{24q_{SD}+48}$, and the adversary $\mathcal{A}_{\mathsf{Sig}}$ as running $\mathcal{A}^0_{\mathsf{Sig}}$ with probability $\frac{4}{24}$, $\mathcal{A}^1_{\mathsf{Sig}}$ with probability $\frac{12}{24}$, and $\mathcal{A}^2_{\mathsf{Sig}}$ with probability $\frac{8}{24}$. Using the result above, we finally conclude

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{sPKE\text{-}SAN\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}} \leq\ & 8 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{g,\mathcal{A}_{\mathsf{DDH}}} + (24q_{SD} + 48) \cdot \mathsf{Adv}^{\mathsf{NIZK\text{-}snd}}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{snd}}} \\
&+ 24 \cdot \mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}}} + \frac{52q_G^2 + 192q_G + 196}{2^\kappa - 1}.
\end{aligned}$$

□

6 Construction of an ACE Scheme

6.1 Construction for Equality

Following Fuchsbauer et al. [FGKO17], we first construct an ACE scheme for the equality policy, i.e., $P(i, j) = 1 \Leftrightarrow i = j$, and then use such a scheme in another construction for richer policies. We base our construction on an sPKE scheme, which already has many important properties needed for a secure ACE scheme. A syntactical difference between sPKE and ACE schemes is that the key generation of the former on every invocation produces a fresh key pair, while the latter schemes allow the generation of keys for a given role. To bind key pairs to some role $i \in [n]$, we use the output of a pseudorandom function on input i as the randomness for the sPKE key generation. For role-respecting security, we have to ensure that an adversary can only produce ciphertexts for keys obtained from the key generation oracle. This is achieved by signing all keys with a signing key generated at setup. To prevent malleability attacks as the ones described in Section 3, the encryption algorithm additionally signs all ciphertexts with a separate signing key that is tied to the encryption key. To maintain anonymity, the signatures are not part of the

ciphertext but the encrypters prove in zero-knowledge that they know such signatures. Finally, the modification detection simply checks whether the ciphertexts (without the NIZK proofs) are equal. Intuitively, this is sufficient since we assume the underlying sPKE scheme to be CCA secure, which implies that it is not possible to meaningfully modify a given ciphertext. Hence, a ciphertext is either equal to an existing one (and thus detected by the algorithm) or a fresh encryption.

Our construction. Let sPKE be a sanitizable public-key encryption scheme, let Sig be a signature scheme, and let F be a PRF. Further let NIZK be a NIZK proof of knowledge system for the language $L := \{x \mid \exists w\ (x,w) \in R\}$, where the relation R is defined as follows: for $x = \big(vk^{\mathsf{Sig}}, \tilde{c}\big)$ and $w = \big(ek_i^{\mathsf{sPKE}}, m, r, vk_i^{\mathsf{Sig}}, \sigma_i^{\mathsf{Sig}}, \sigma_c^{\mathsf{Sig}}\big)$, $(x,w) \in R$ if and only if

$$\tilde{c} = \mathsf{sPKE.Enc}\big(ek_i^{\mathsf{sPKE}}, m; r\big) \ \wedge\ \mathsf{Sig.Ver}\big(vk^{\mathsf{Sig}}, \big[ek_i^{\mathsf{sPKE}}, vk_i^{\mathsf{Sig}}\big], \sigma_i^{\mathsf{Sig}}\big) = 1$$
$$\wedge\ \mathsf{Sig.Ver}\big(vk_i^{\mathsf{Sig}}, \tilde{c}, \sigma_c^{\mathsf{Sig}}\big) = 1.$$

We define an ACE with modification detection scheme ACE as follows:

Setup: On input a security parameter 1^κ and a policy $P\colon [n] \times [n] \to \{0,1\}$ with $P(i,j) = 1 \Leftrightarrow i = j$, the algorithm $\mathsf{ACE.Setup}$ picks a random PRF key K for a PRF F, and runs

$$\big(sp^{\mathsf{sPKE}}, msk^{\mathsf{sPKE}}\big) \leftarrow \mathsf{sPKE.Setup}(1^\kappa),$$
$$\big(vk^{\mathsf{Sig}}, sk^{\mathsf{Sig}}\big) \leftarrow \mathsf{Sig.Gen}(1^\kappa),$$
$$crs^{\mathsf{NIZK}} \leftarrow \mathsf{NIZK.Gen}(1^\kappa).$$

It outputs the master secret key $msk^{\mathsf{ACE}} := \big(K, msk^{\mathsf{sPKE}}, vk^{\mathsf{Sig}}, sk^{\mathsf{Sig}}, crs^{\mathsf{NIZK}}\big)$ and the sanitizer parameters $sp^{\mathsf{ACE}} := \big(sp^{\mathsf{sPKE}}, vk^{\mathsf{Sig}}, crs^{\mathsf{NIZK}}\big)$.

Key generation: The algorithm $\mathsf{ACE.Gen}$ on input a master secret key $msk^{\mathsf{ACE}} = \big(K, msk^{\mathsf{sPKE}}, vk^{\mathsf{Sig}}, sk^{\mathsf{Sig}}, crs^{\mathsf{NIZK}}\big)$, a role $i \in [n]$, and a type $t \in \{\mathtt{sen}, \mathtt{rec}\}$, computes

$$\big(ek_i^{\mathsf{sPKE}}, dk_i^{\mathsf{sPKE}}\big) \leftarrow \mathsf{sPKE.Gen}\big(msk^{\mathsf{sPKE}}; F_K([i,0])\big).$$

If $t = \mathtt{sen}$, it further computes

$$\big(vk_i^{\mathsf{Sig}}, sk_i^{\mathsf{Sig}}\big) \leftarrow \mathsf{Sig.Gen}\big(1^\kappa; F_K([i,1])\big),$$
$$\sigma_i^{\mathsf{Sig}} \leftarrow \mathsf{Sig.Sign}\big(sk^{\mathsf{Sig}}, \big[ek_i^{\mathsf{sPKE}}, vk_i^{\mathsf{Sig}}\big]; F_K([i,2])\big).$$

If $t = \mathtt{sen}$, it outputs the encryption key $ek_i^{\mathsf{ACE}} := \big(vk^{\mathsf{Sig}}, ek_i^{\mathsf{sPKE}}, vk_i^{\mathsf{Sig}}, sk_i^{\mathsf{Sig}}, \sigma_i^{\mathsf{Sig}}, crs^{\mathsf{NIZK}}\big)$; if $t = \mathtt{rec}$, it outputs the decryption key $dk_i^{\mathsf{ACE}} := dk_i^{\mathsf{sPKE}}$.

Encryption: On input an encryption key $ek_i^{\mathsf{ACE}} = \big(vk^{\mathsf{Sig}}, ek_i^{\mathsf{sPKE}}, vk_i^{\mathsf{Sig}}, sk_i^{\mathsf{Sig}}, \sigma_i^{\mathsf{Sig}}, crs^{\mathsf{NIZK}}\big)$ and a message $m \in \mathcal{M}^{\mathsf{ACE}}$, the algorithm $\mathsf{ACE.Enc}$ samples randomness r and computes

$$\tilde{c} \leftarrow \mathsf{sPKE.Enc}\big(ek_i^{\mathsf{sPKE}}, m; r\big),$$
$$\sigma_c^{\mathsf{Sig}} \leftarrow \mathsf{Sig.Sign}\big(sk_i^{\mathsf{Sig}}, \tilde{c}\big),$$
$$\pi^{\mathsf{NIZK}} \leftarrow \mathsf{NIZK.Prove}\big(crs^{\mathsf{NIZK}}, x := \big(vk^{\mathsf{Sig}}, \tilde{c}\big), w := \big(ek_i^{\mathsf{sPKE}}, m, r, vk_i^{\mathsf{Sig}}, \sigma_i^{\mathsf{Sig}}, \sigma_c^{\mathsf{Sig}}\big)\big).$$

It outputs the ciphertext $c := \big(\tilde{c}, \pi^{\mathsf{NIZK}}\big)$.

Sanitization: On input sanitizer parameters $sp^{\mathsf{ACE}} = \big(sp^{\mathsf{sPKE}}, vk^{\mathsf{Sig}}, crs^{\mathsf{NIZK}}\big)$ and a ciphertext $c = \big(\tilde{c}, \pi^{\mathsf{NIZK}}\big)$, $\mathsf{ACE.San}$ outputs the sanitized ciphertext $c' \leftarrow \mathsf{sPKE.San}\big(sp^{\mathsf{sPKE}}, \tilde{c}\big)$ if $\mathsf{NIZK.Ver}\big(crs^{\mathsf{NIZK}}, x \coloneqq \big(vk^{\mathsf{Sig}}, \tilde{c}, \big), \pi^{\mathsf{NIZK}}\big) = 1$; otherwise, it outputs $\perp$.

Decryption: The algorithm $\mathsf{ACE.Dec}$ on input a decryption key dk_j^{ACE} and a sanitized ciphertext c', outputs the message $m \leftarrow \mathsf{sPKE.Dec}(dk_j^{\mathsf{ACE}}, c')$.

Modification detection: The algorithm $\mathsf{ACE.DMod}$ on input sp^{ACE}, $c_1 = \big(\tilde{c}_1, \pi_1^{\mathsf{NIZK}}\big)$, and $c_2 = \big(\tilde{c}_2, \pi_2^{\mathsf{NIZK}}\big)$, outputs 1 if $\tilde{c}_1 = \tilde{c}_2$, and 0 otherwise.

We first show that our scheme is correct and strongly detectable.

Proposition 6.1. *Let* ACE *be the scheme from above. Then,* ACE *is perfectly correct, i.e.,* $\mathsf{Adv}^{\mathsf{ACE\text{-}CORR}}_{\mathsf{ACE},\mathcal{A}} = 0$ *for all* $\mathcal{A}$*. Moreover, if* F *is pseudorandom and* sPKE *is unrestricted strongly robust, then* ACE *is strongly detectable.*

Proof. Perfect correctness follows from the perfect correctness of the sPKE and signature schemes and the perfect completeness of the NIZK proof system.

To prove strong detectability, let $\mathcal{A}$ be a probabilistic algorithm. We assume without loss of generality that $\mathcal{A}$ returns (m, r, i, j) with $P(i, j) = 0$ since doing otherwise can only reduce the advantage. Let $H_0 \coloneqq \mathsf{Exp}^{\mathsf{ACE\text{-}sDTCT}}_{\mathsf{ACE},\mathcal{A}}$, let H_1 be as H_0 where F_K is replaced by a truly uniform function U, and let W be the event that $\mathcal{A}$ wins the strong detectability game, i.e.,

$$W \coloneqq \big[\mathsf{ACE.Dec}\big(dk_j^{\mathsf{ACE}}, \mathsf{ACE.San}(sp^{\mathsf{ACE}}, \mathsf{ACE.Enc}(ek_i^{\mathsf{ACE}}, m; r))\big) \neq \perp\big].$$

We first show that the difference in the winning probability in H_0 and H_1 is bounded by the PRF advantage.

Claim 1. *There exists a probabilistic algorithm* $\mathcal{A}^{\mathcal{O}(\cdot)}_{\mathsf{PRF}}$ *such that*

$$\Pr^{H_0}[W] - \Pr^{H_1}[W] = \mathsf{Adv}^{\mathsf{PRF}}_{F,\mathcal{A}_{\mathsf{PRF}}}.$$

Proof of claim. Consider $\mathcal{A}^{\mathcal{O}(\cdot)}_{\mathsf{PRF}}$ that emulates an execution of H_0, where all invocations of $F_K(\cdot)$ are replaced by a call to the oracle $\mathcal{O}(\cdot)$. When $\mathcal{A}$ wins, $\mathcal{A}_{\mathsf{PRF}}$ outputs 1, and 0 otherwise. In case $\mathcal{O}(\cdot)$ corresponds to $F_K(\cdot)$, $\mathcal{A}_{\mathsf{PRF}}$ perfectly emulates H_0, if it corresponds to $U(\cdot)$, it perfectly emulates H_1. Hence,

$$\Pr^{H_0}[W] - \Pr^{H_1}[W] = \Pr\Big[\mathcal{A}^{F_K(\cdot)}_{\mathsf{PRF}}(1^\kappa) = 1\Big] - \Pr\Big[\mathcal{A}^{U(\cdot)}_{\mathsf{PRF}}(1^\kappa) = 1\Big] = \mathsf{Adv}^{\mathsf{PRF}}_{F,\mathcal{A}_{\mathsf{PRF}}}. \qquad \Diamond$$

We now construct a winner $\mathcal{A}_{\mathsf{rob}}$ for the robustness game for sPKE. The algorithm $\mathcal{A}_{\mathsf{rob}}$ on input sp^{sPKE} emulates an execution of H_1. To answer queries of $\mathcal{A}$ to the key-generation oracle, $\mathcal{A}_{\mathsf{rob}}$ uses the oracle $\mathcal{O}_G$ to obtain encryption and decryption keys for sPKE; the required signature keys are generated internally. For each query (i, t), $\mathcal{A}_{\mathsf{rob}}$ remembers the generated keys ek_i^{ACE} and dk_i^{ACE}, and returns the same keys for subsequent queries with the same i. When $\mathcal{A}$ returns (m, r, i, j), $\mathcal{A}_{\mathsf{rob}}$ first checks whether i and j have been queried by $\mathcal{A}$ to the key-generation oracle. If not, $\mathcal{A}_{\mathsf{rob}}$ now generates these keys as above. Let $\tilde{r}$ be the randomness used by $\mathsf{ACE.Enc}(ek_i^{\mathsf{ACE}}, m; r))$ for the algorithm $\mathsf{sPKE.Enc}$. Then, $\mathcal{A}_{\mathsf{rob}}$ computes $c \leftarrow \mathsf{sPKE.Enc}\big(ek_i^{\mathsf{sPKE}}, m; \tilde{r}\big)$, and returns (c, i_0, i_1), such that the i_0-th query and the i_1-th query to the key-generation oracle were for the roles i and j, respectively. Since P is the equality

predicate, $P(i,j)=0$ is equivalent to $i_0 \neq i_1$. We further have by the perfect correctness of sPKE that $\mathsf{sPKE.Dec}\big(dk_i^{\mathsf{sPKE}}, \mathsf{sPKE.San}\big(sp^{\mathsf{sPKE}}, c\big)\big) \neq \bot$. Hence, $\mathcal{A}_{\mathsf{rob}}$ wins the robustness game if and only if $\mathcal{A}$ wins the strong detectability game in H_1. Using Claim 1, we can therefore conclude

$$\mathsf{Adv}^{\mathsf{ACE\text{-}sDTCT}}_{\mathsf{ACE},\mathcal{A}} = \Pr^{H_0}[W] = \mathsf{Adv}^{\mathsf{PRF}}_{F,\mathcal{A}_{\mathsf{PRF}}} + \Pr^{H_1}[W] = \mathsf{Adv}^{\mathsf{PRF}}_{F,\mathcal{A}_{\mathsf{PRF}}} + \mathsf{Adv}^{\mathsf{sPKE\text{-}USROB}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{rob}}}. \qquad \square$$

In the following, we prove the security of our scheme, which is summarized by the theorem below.

Theorem 6.2. *If F is pseudorandom, NIZK is zero-knowledge and extractable, Sig is EUF-CMA secure, and sPKE is IND-CCA, IK-CCA, SAN-CCA, USROB, and UPD-CTXT secure and has negligible encryption-key collision probability, then the scheme ACE from above is PRV-CCA, sANON-CCA, SAN-CCA, UDEC, and RR secure, and has NDTCT-FENC.*

We first show that our scheme satisfies the privacy definition from Definition 4.2 if the underlying sanitizable public-key encryption scheme is IND-CCA secure, the PRF is secure, and the NIZK is zero-knowledge.

Lemma 6.3. *Let ACE be the scheme from above, let $\mathcal{A}=(\mathcal{A}_1,\mathcal{A}_2)$ be an attacker on the privacy such that $\mathcal{A}_1$ makes at most q_S queries of the form $(\cdot,\mathtt{sen})$ to the oracle $\mathcal{O}_G$, and at most q_D queries to $\mathcal{O}_{SD}$. Then, there exist probabilistic algorithms $\mathcal{A}_{\mathsf{PRF}}$, $\mathcal{A}_{\mathsf{ZK}}$, and $\mathcal{A}_{\mathsf{sPKE}}$ (which are all roughly as efficient as emulating an execution of $\mathsf{Exp}^{\mathsf{ACE\text{-}PRV\text{-}ANON\text{-}CCA}}_{\mathsf{ACE},\mathcal{A}}$) such that*

$$\mathsf{Adv}^{\mathsf{ACE\text{-}PRV\text{-}CCA}}_{\mathsf{ACE},\mathcal{A}} \leq 2\cdot \mathsf{Adv}^{\mathsf{PRF}}_{F,\mathcal{A}_{\mathsf{PRF}}} + 2\cdot \mathsf{Adv}^{\mathsf{NIZK\text{-}ZK}}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}}} + (q_S+q_D+1)\cdot \mathsf{Adv}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}.$$

Proof. We assume without loss of generality that $\mathcal{A}$ ensures $i_0=i_1$ and $P(i_0,j)=0$ for all $j\in J$, since doing otherwise can only decrease the advantage. Let $H_0 := \mathsf{Exp}^{\mathsf{ACE\text{-}PRV\text{-}ANON\text{-}CCA}}_{\mathsf{ACE},\mathcal{A}}$ and H_1 be as H_0 where F_K is replaced by a truly uniform random function U. The following can be proven as Claim 1 in the proof of Proposition 6.1.

Claim 1. *There exists a probabilistic algorithm $\mathcal{A}^{\mathcal{O}(\cdot)}_{\mathsf{PRF}}$ such that*

$$\Pr^{H_0}[b'=b] - \Pr^{H_1}[b'=b] = \mathsf{Adv}^{\mathsf{PRF}}_{F,\mathcal{A}_{\mathsf{PRF}}}.$$

Now let H_2 be as H_1, where we replace $crs^{\mathsf{NIZK}} \leftarrow \mathsf{NIZK.Gen}(1^\kappa)$ by $\big(crs^{\mathsf{NIZK}}, \tau^{\mathsf{NIZK}}\big) \leftarrow S_1^{\mathsf{NIZK}}(1^\kappa)$ in $\mathsf{ACE.Setup}$, and for the generation of the challenge ciphertext c^*, we replace $\pi^{\mathsf{NIZK}} \leftarrow \mathsf{NIZK.Prove}\big(crs^{\mathsf{NIZK}}, x, w\big)$ in $\mathsf{ACE.Enc}$ by $\pi^{\mathsf{NIZK}} \leftarrow S_2^{\mathsf{NIZK}}\big(crs^{\mathsf{NIZK}}, \tau^{\mathsf{NIZK}}, x\big)$.

Claim 2. *There exists a probabilistic algorithm $\mathcal{A}^{\mathcal{O}(\cdot,\cdot)}_{\mathsf{ZK}}$ such that*

$$\Pr^{H_1}[b'=b] - \Pr^{H_2}[b'=b] = \mathsf{Adv}^{\mathsf{NIZK\text{-}ZK}}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}}}.$$

Proof of claim. The algorithm $\mathcal{A}^{\mathcal{O}(\cdot,\cdot)}_{\mathsf{ZK}}$ on input crs^{NIZK} proceeds as follows. It emulates an execution of H_1, where in $\mathsf{ACE.Setup}$, crs^{NIZK} is used instead of generating it, and for the generation of c^*, $\mathsf{NIZK.Prove}\big(crs^{\mathsf{NIZK}}, x, w\big)$ in $\mathsf{ACE.Enc}$ is replaced by the oracle query (x,w). Finally, $\mathcal{A}^{\mathcal{O}(\cdot,\cdot)}_{\mathsf{ZK}}$ outputs $\tilde{b}=1$ if $\mathcal{A}_2$ returns $b'=b$, and $\tilde{b}=0$ otherwise. Note that if crs^{NIZK} is generated by $\mathsf{NIZK.Gen}$ and $\mathcal{O}(\cdot,\cdot)$ corresponds to $\mathsf{NIZK.Prove}\big(crs^{\mathsf{NIZK}},\cdot,\cdot\big)$, $\mathcal{A}^{\mathcal{O}(\cdot,\cdot)}_{\mathsf{ZK}}$ perfectly emulates H_1. Moreover, if crs^{NIZK} is generated together with τ^{NIZK} by S_1^{NIZK} and $\mathcal{O}(x,w)$ returns $S_2^{\mathsf{NIZK}}\big(crs^{\mathsf{NIZK}}, \tau^{\mathsf{NIZK}}, x\big)$, $\mathcal{A}^{\mathcal{O}(\cdot,\cdot)}_{\mathsf{ZK}}$ perfectly emulates H_2. Thus, the claim follows. $\lozenge$

We finally show how to transform any winner $\mathcal{A}$ for H_2 to a winner $\mathcal{A}_{\mathsf{sPKE}}$ for the IND-CCA game for the scheme sPKE. The strategy of our reduction is to guess which oracle queries of $\mathcal{A}_1$ are for the role i_0, use the key from the sPKE-scheme for these queries, and generate all other keys as H_2. Details follow. On input $(sp^{\mathsf{sPKE}}, ek^{\mathsf{sPKE}})$, $\mathcal{A}_{\mathsf{sPKE}}$ initializes $i_{q_0} \leftarrow \perp$, $k_q \leftarrow 1$, chooses $q_0 \twoheadleftarrow \{0, \ldots, q_S + q_D\}$ uniformly at random, runs $\big(vk^{\mathsf{Sig}}, sk^{\mathsf{Sig}}\big) \leftarrow \mathsf{Sig.Gen}(1^\kappa)$, and $\big(crs^{\mathsf{NIZK}}, \tau^{\mathsf{NIZK}}\big) \leftarrow S_1^{\mathsf{NIZK}}(1^\kappa)$, and gives $sp^{\mathsf{ACE}} := \big(sp^{\mathsf{sPKE}}, vk^{\mathsf{Sig}}, crs^{\mathsf{NIZK}}\big)$ to $\mathcal{A}_1$. It emulates the oracles for $\mathcal{A}_1$ as follows.

$\mathcal{O}_G(\cdot,\cdot)$**:** On query $(i, \mathtt{sen})$, if $k_q \neq q_0$ and $i \neq i_{q_0}$, then generate an encryption key $ek_i^{\mathsf{ACE}} := \big(vk^{\mathsf{Sig}}, ek_i^{\mathsf{sPKE}}, vk_i^{\mathsf{Sig}}, sk_i^{\mathsf{Sig}}, \sigma_i^{\mathsf{Sig}}, crs^{\mathsf{NIZK}}\big)$ as H_2 does, where $\big(ek_i^{\mathsf{sPKE}}, dk_i^{\mathsf{sPKE}}\big)$ is obtained via $\mathcal{O}_G$ and remembered for future queries. If $k_q = q_0$ or $i = i_{q_0}$, replace ek_i^{sPKE} by ek^{sPKE} and set $i_{q_0} \leftarrow i$. In both cases, set $k_q \leftarrow k_q + 1$ at the end. On query $(j, \mathtt{rec})$, obtain a decryption key via $\mathcal{O}_G$.

$\mathcal{O}_{SD}(\cdot,\cdot)$**:** On query $\big(j, c = (\tilde{c}, \pi^{\mathsf{NIZK}})\big)$, if $k_q \neq q_0$ and $j \neq i_{q_0}$, run $c' \leftarrow \mathsf{ACE.San}(sp^{\mathsf{ACE}}, c)$, generate a decryption key dk_j^{ACE} as above, decrypt c' using dk_j^{ACE}, and return the resulting message. If $k_q = q_0$ or $j = i_{q_0}$, set $i_{q_0} \leftarrow j$ and use the oracle $\mathcal{O}_{SD}$ of the IND-CCA experiment to obtain a decryption m of $\tilde{c}$. If $\mathsf{NIZK.Ver}\big(crs^{\mathsf{NIZK}}, x := \big(vk^{\mathsf{Sig}}, \tilde{c}, \big), \pi^{\mathsf{NIZK}}\big) = 1$, return m, otherwise, return $\perp$. In all cases, set $k_q \leftarrow k_q + 1$ at the end.

When $\mathcal{A}_1$ returns (m_0, m_1, i_0, i_1, st), output (m_0, m_1) to the challenger of the IND-CCA experiment to obtain a challenge ciphertext $\tilde{c}^*$. Then run $\pi^{\mathsf{NIZK}} \leftarrow S_2^{\mathsf{NIZK}}\big(crs^{\mathsf{NIZK}}, \tau^{\mathsf{NIZK}}, x := \big(vk^{\mathsf{Sig}}, \tilde{c}^*\big)\big)$, and give st and the ciphertext $c^* := \big(\tilde{c}^*, \pi^{\mathsf{NIZK}}\big)$ to $\mathcal{A}_2$. Emulate the oracles for $\mathcal{A}_2$ as follows.

$\mathcal{O}_G(\cdot,\cdot)$**:** On query $(i, \mathtt{sen})$, if $i \neq i_0$, then generate an encryption key $ek_i^{\mathsf{ACE}} := \big(vk^{\mathsf{Sig}}, ek_i^{\mathsf{sPKE}}, vk_i^{\mathsf{Sig}}, sk_i^{\mathsf{Sig}}, \sigma_i^{\mathsf{Sig}}, crs^{\mathsf{NIZK}}\big)$ as H_2 does, where $\big(ek_i^{\mathsf{sPKE}}, dk_i^{\mathsf{sPKE}}\big)$ is obtained via $\mathcal{O}_G$ and remembered for future queries. If $i = i_0$, replace ek_i^{sPKE} by ek^{sPKE}. On query $(j, \mathtt{rec})$, obtain a decryption key from $\mathcal{O}_G$.

$\mathcal{O}_{SD^*}(\cdot,\cdot)$**:** On query $\big(j, c = (\tilde{c}, \pi^{\mathsf{NIZK}})\big)$, run $\mathsf{ACE.DMod}(sp^{\mathsf{ACE}}, c^*, c)$. If the output is 1, return $\mathtt{test}$. Otherwise, if $j \neq i_0$, run $c' \leftarrow \mathsf{ACE.San}(sp^{\mathsf{ACE}}, c)$, generate a decryption key dk_j^{ACE} as above, decrypt c' using dk_j^{ACE}, and return the resulting message. If $j = i_0$, use the oracle $\mathcal{O}_{SD}$ of the IND-CCA experiment to obtain a decryption m of $\tilde{c}$. If $\mathsf{NIZK.Ver}\big(crs^{\mathsf{NIZK}}, x := \big(vk^{\mathsf{Sig}}, \tilde{c}, \big), \pi^{\mathsf{NIZK}}\big) = 1$, return m, otherwise, return $\perp$.

Note that we never query the decryption oracle of the IND-CCA experiment on $\tilde{c}^*$ because we return $\mathtt{test}$ whenever this would be necessary. Denote by Q the event that either $i_{q_0} = i_0$, or $q_0 = 0$ and $\mathcal{A}_1$ does not make the query $(i_0, \mathtt{sen})$ to $\mathcal{O}_G$ and no queries for role i_0 to $\mathcal{O}_{SD}$. When $\mathcal{A}_2$ returns a bit b' and Q holds, $\mathcal{A}_{\mathsf{sPKE}}$ returns the same bit $b'' \leftarrow b'$, if $\neg Q$, $\mathcal{A}_{\mathsf{sPKE}}$ returns a uniform bit $b'' \twoheadleftarrow \{0,1\}$.

Let $\tilde{b}$ be the bit chosen by the IND-CCA challenger. Note that by our assumption on $\mathcal{A}$, $i_0 = i_1$ and $\mathcal{A}$ does not query $(i_0, \mathtt{rec})$ to $\mathcal{O}_G$, i.e., $i_0 \notin J$, since $P(i_0, i_0) = 1$. Hence, if Q occurs, the view of $\mathcal{A}$ is identical to the one in H_2 with $b = \tilde{b}$. This implies

$$\Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}\big[b'' = \tilde{b} \;\big|\; Q\big] = \Pr^{H_2}\big[b' = b\big],$$

and therefore

$$\begin{aligned}\Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}\left[b''=\tilde{b}\right] &= \Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}\left[b''=\tilde{b}\mid Q\right]\cdot\Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}[Q]\\ &\quad+\Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}\left[b''=\tilde{b}\mid \neg Q\right]\cdot\Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}[\neg Q]\\ &= \Pr^{H_2}\left[b'=b\right]\cdot\Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}[Q]+\frac{1}{2}\Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}[\neg Q].\end{aligned}$$

Using that the probability of Q is $1/(q_S+q_D+1)$, this yields

$$\begin{aligned}&\Pr^{H_2}\left[b'=b\right]\\ &=\frac{1}{\Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}[Q]}\cdot\left(\Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}\left[b''=\tilde{b}\right]-\frac{1}{2}\cdot\left(1-\Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}[Q]\right)\right)\\ &=(q_S+q_D+1)\cdot\left(\Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}\left[b''=\tilde{b}\right]-\frac{1}{2}\right)+\frac{1}{2}.\end{aligned}$$

Combining this with Claims 1 and 2, we can conclude

$$\begin{aligned}&\mathsf{Adv}^{\mathsf{ACE\text{-}PRV\text{-}CCA}}_{\mathsf{ACE},\mathcal{A}}\\ &=2\cdot\Pr^{H_0}[b'=b]-1\\ &=2\cdot\left(\Pr^{H_0}[b'=b]-\Pr^{H_1}[b'=b]+\Pr^{H_1}[b'=b]-\Pr^{H_2}[b'=b]+\Pr^{H_2}[b'=b]\right)-1\\ &=2\cdot\left[\mathsf{Adv}^{\mathsf{PRF}}_{F,\mathcal{A}_{\mathsf{PRF}}}+\mathsf{Adv}^{\mathsf{NIZK\text{-}ZK}}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}}}+(q_S+q_D+1)\left(\Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}\left[b''=\tilde{b}\right]-\frac{1}{2}\right)+\frac{1}{2}\right]-1\\ &=2\cdot\mathsf{Adv}^{\mathsf{PRF}}_{F,\mathcal{A}_{\mathsf{PRF}}}+2\cdot\mathsf{Adv}^{\mathsf{NIZK\text{-}ZK}}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}}}+(q_S+q_D+1)\cdot\mathsf{Adv}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}.\end{aligned}$$

□

The proofs of the other properties use similar techniques and can be found in Appendix D.

6.2 Lifting Equality to Disjunction of Equalities

We finally show how an ACE scheme for equality, as the one from Section 6.1, can be used to construct a scheme for the policy $P_{\mathsf{DEq}}\colon \mathcal{D}^\ell\times\mathcal{D}^\ell\to\{0,1\}$ with

$$P_{\mathsf{DEq}}(\mathbf{x}=(x_1,\ldots,x_\ell),\mathbf{y}=(y_1,\ldots,y_\ell))=1\ :\Longleftrightarrow\ \bigvee_{i=1}^{\ell}x_i=y_i,$$

where $\mathcal{D}$ is some finite set and $\ell\in\mathbb{N}$.[4] This policy can for example be used to implement the no read-up and now write-down principle ($P(i,j)=1\Leftrightarrow i\le j$) from the Bell–LaPadula model [BL73] via an appropriate encoding of the roles [FGKO17].

The intuition of our construction is as follows. A key for a role $\mathbf{x}=(x_1,\ldots,x_\ell)$ contains one key of the ACE scheme for equality for each component x_i of the role vector. To encrypt a message, this message is encrypted with each of these keys. To decrypt, one tries to decrypt each ciphertext component with the corresponding key. If at least one component of the sender and receiver roles match (i.e., if the policy is satisfied), one of the decryptions is successful. So

[4] In this section, we denote roles by $\mathbf{x}$ and $\mathbf{y}$ instead of i and j. To be compatible with our definitions that consider policies $[n]\times[n]\to\{0,1\}$, one needs to identify elements of $\mathcal{D}^\ell$ with numbers in $[n]$. We will ignore this technicality to simplify the presentation.

far, the construction is identical to the one by Fuchsbauer et al. [FGKO17]. That construction is, however, not role-respecting, since a dishonest sender with keys for more than one role can arbitrarily mix the components of the keys for the encryption. Moreover, the construction does not guarantee uniform decryption, because different messages can be encrypted in different components. We fix these issues using the same techniques we used in our construction of the scheme for equality, i.e., we add a signature of the key vector to the encryption keys, sign the ciphertexts, and require a zero-knowledge proof of knowledge that a valid key combination was used to encrypt the same message for each component and that all signatures are valid.

Our construction. Let $\mathsf{ACE}_{=}$ be an ACE with modification detection scheme for the equality predicate on $\mathcal{D} \times [\ell]$, let Sig be a signature scheme, let F be a PRF, and let NIZK be a NIZK proof of knowledge system for the language $L := \{x \mid \exists w\ (x, w) \in R\}$, where the relation R is defined as follows: for $x = \big(vk^{\mathsf{Sig}}, c_1, \ldots, c_\ell\big)$ and $w = \big(ek^{\mathsf{ACE}_=}_{(x_1,1)}, \ldots, ek^{\mathsf{ACE}_=}_{(x_\ell,\ell)}, m, r_1, \ldots, r_\ell, vk^{\mathsf{Sig}}_{\mathbf{x}}, \sigma^{\mathsf{Sig}}_{\mathbf{x}}, \sigma^{\mathsf{Sig}}_{c}\big)$, $(x, w) \in R$ if and only if

$$\bigwedge_{i=1}^{\ell} c_i = \mathsf{ACE}_{=}.\mathsf{Enc}\big(ek^{\mathsf{ACE}_=}_{(x_i,i)}, m; r_i\big) \ \wedge\ \mathsf{Sig}.\mathsf{Ver}\big(vk^{\mathsf{Sig}}_{\mathbf{x}}, [c_1, \ldots, c_\ell], \sigma^{\mathsf{Sig}}_{c}\big) = 1$$
$$\wedge\ \mathsf{Sig}.\mathsf{Ver}\big(vk^{\mathsf{Sig}}, \big[ek^{\mathsf{ACE}_=}_{(x_1,1)}, \ldots, ek^{\mathsf{ACE}_=}_{(x_\ell,\ell)}, vk^{\mathsf{Sig}}_{\mathbf{x}}\big], \sigma^{\mathsf{Sig}}_{\mathbf{x}}\big) = 1.$$

We define an ACE scheme $\mathsf{ACE}_{\mathsf{DEq}}$ as follows:

Setup: On input a security parameter 1^κ and the policy P_{DEq}, the algorithm $\mathsf{ACE}_{\mathsf{DEq}}.\mathsf{Setup}$ picks a random key K for F and runs

$$\big(msk^{\mathsf{ACE}_=}, sp^{\mathsf{ACE}_=}\big) \leftarrow \mathsf{ACE}_{=}.\mathsf{Setup}(1^\kappa),$$
$$\big(vk^{\mathsf{Sig}}, sk^{\mathsf{Sig}}\big) \leftarrow \mathsf{Sig}.\mathsf{Gen}(1^\kappa),$$
$$crs^{\mathsf{NIZK}} \leftarrow \mathsf{NIZK}.\mathsf{Gen}(1^\kappa).$$

It outputs the master secret key $msk^{\mathsf{ACE}_{\mathsf{DEq}}} := \big(K, msk^{\mathsf{ACE}_=}, vk^{\mathsf{Sig}}, sk^{\mathsf{Sig}}, crs^{\mathsf{NIZK}}\big)$ and the sanitizer parameters $sp^{\mathsf{ACE}_{\mathsf{DEq}}} := \big(sp^{\mathsf{ACE}_=}, vk^{\mathsf{Sig}}, crs^{\mathsf{NIZK}}\big)$.

Key generation: The algorithm $\mathsf{ACE}_{\mathsf{DEq}}.\mathsf{Gen}$ on input a master secret key $msk^{\mathsf{ACE}_{\mathsf{DEq}}} = \big(K, msk^{\mathsf{ACE}_=}, vk^{\mathsf{Sig}}, sk^{\mathsf{Sig}}, crs^{\mathsf{NIZK}}\big)$, a role $\mathbf{x} \in \mathcal{D}^\ell$, and the type $\mathtt{sen}$, generates

$$ek^{\mathsf{ACE}_=}_{(x_i,i)} \leftarrow \mathsf{ACE}_{=}.\mathsf{Gen}\big(msk^{\mathsf{ACE}_=}, (x_i, i), \mathtt{sen}\big) \quad (\text{for } i \in [\ell]),$$
$$\big(vk^{\mathsf{Sig}}_{\mathbf{x}}, sk^{\mathsf{Sig}}_{\mathbf{x}}\big) \leftarrow \mathsf{Sig}.\mathsf{Gen}(1^\kappa; F_K([\mathbf{x}, 0])),$$
$$\sigma^{\mathsf{Sig}}_{\mathbf{x}} \leftarrow \mathsf{Sig}.\mathsf{Sign}\big(sk^{\mathsf{Sig}}, \big[ek^{\mathsf{ACE}_=}_{(x_1,1)}, \ldots, ek^{\mathsf{ACE}_=}_{(x_\ell,\ell)}, vk^{\mathsf{Sig}}_{\mathbf{x}}\big]; F_K([\mathbf{x}, 1])\big),$$

and outputs the encryption key $ek^{\mathsf{ACE}_{\mathsf{DEq}}}_{\mathbf{x}} := \big(vk^{\mathsf{Sig}}, ek^{\mathsf{ACE}_=}_{(x_1,1)}, \ldots, ek^{\mathsf{ACE}_=}_{(x_\ell,\ell)}, vk^{\mathsf{Sig}}_{\mathbf{x}}, sk^{\mathsf{Sig}}_{\mathbf{x}}, \sigma^{\mathsf{Sig}}_{\mathbf{x}}, crs^{\mathsf{NIZK}}\big)$; on input $msk^{\mathsf{ACE}_{\mathsf{DEq}}}$, a role $\mathbf{y} \in \mathcal{D}^\ell$, and the type $\mathtt{rec}$, it generates for $i \in [\ell]$,

$$dk^{\mathsf{ACE}_=}_{(y_i,i)} \leftarrow \mathsf{ACE}_{=}.\mathsf{Gen}\big(msk^{\mathsf{ACE}_=}, (y_i, i), \mathtt{rec}\big),$$

and outputs the decryption key $dk^{\mathsf{ACE}_{\mathsf{DEq}}}_{\mathbf{y}} := \big(dk^{\mathsf{ACE}_=}_{(y_1,1)}, \ldots, dk^{\mathsf{ACE}_=}_{(y_\ell,\ell)}\big)$.

Encryption: On input an encryption key $ek_{\mathbf{x}}^{\mathsf{ACE}_{\mathsf{DEq}}} = \big(vk^{\mathsf{Sig}}, ek_{(x_1,1)}^{\mathsf{ACE}_=}, \ldots, ek_{(x_\ell,\ell)}^{\mathsf{ACE}_=}, vk_{\mathbf{x}}^{\mathsf{Sig}}, sk_{\mathbf{x}}^{\mathsf{Sig}}, \sigma_{\mathbf{x}}^{\mathsf{Sig}}, crs^{\mathsf{NIZK}}\big)$ and a message $m \in \mathcal{M}^{\mathsf{ACE}_{\mathsf{DEq}}}$, the algorithm $\mathsf{ACE}_{\mathsf{DEq}}.\mathsf{Enc}$ samples randomness $r_1, \ldots, r_\ell$ and computes

$$\begin{aligned} c_i &\leftarrow \mathsf{ACE}_=.\mathsf{Enc}\big(ek_{(x_i,i)}^{\mathsf{ACE}_=}, m; r_i\big) \quad (\text{for } i \in [\ell]),\\ \sigma_c^{\mathsf{Sig}} &\leftarrow \mathsf{Sig}.\mathsf{Sign}\big(sk_{\mathbf{x}}^{\mathsf{Sig}}, [c_1, \ldots, c_\ell]\big),\\ \pi^{\mathsf{NIZK}} &\leftarrow \mathsf{NIZK}.\mathsf{Prove}\big(crs^{\mathsf{NIZK}}, x \coloneqq \big(vk^{\mathsf{Sig}}, c_1, \ldots, c_\ell\big),\\ &\qquad\qquad w \coloneqq \big(ek_{(x_1,1)}^{\mathsf{ACE}_=}, \ldots, ek_{(x_\ell,\ell)}^{\mathsf{ACE}_=}, m, r_1, \ldots, r_\ell, vk_{\mathbf{x}}^{\mathsf{Sig}}, \sigma_{\mathbf{x}}^{\mathsf{Sig}}, \sigma_c^{\mathsf{Sig}}\big)\big). \end{aligned}$$

It outputs the ciphertext $c \coloneqq \big(c_1, \ldots, c_\ell, \pi^{\mathsf{NIZK}}\big)$.

Sanitization: On input sanitizer parameters $sp^{\mathsf{ACE}_{\mathsf{DEq}}} = \big(sp^{\mathsf{ACE}_=}, vk^{\mathsf{Sig}}, crs^{\mathsf{NIZK}}\big)$ and a ciphertext $c = \big(c_1, \ldots, c_\ell, \pi^{\mathsf{NIZK}}\big)$, the algorithm $\mathsf{ACE}_{\mathsf{DEq}}.\mathsf{San}$ checks whether $\mathsf{NIZK}.\mathsf{Ver}\big(crs^{\mathsf{NIZK}}, x \coloneqq \big(vk^{\mathsf{Sig}}, c_1, \ldots, c_\ell\big), \pi^{\mathsf{NIZK}}\big) = 1$. If this is the case, it runs $c_i' \leftarrow \mathsf{ACE}_=.\mathsf{San}\big(c_i\big)$ for $i \in [\ell]$. If $c_i' \neq \bot$ for all $i \in [\ell]$, it outputs the sanitized ciphertext $c' \coloneqq \big(c_1', \ldots, c_\ell'\big)$. If the verification fails or any of the sanitized ciphertexts is $\bot$, it outputs $\bot$.

Decryption: On input a decryption key $dk_{\mathbf{y}}^{\mathsf{ACE}_{\mathsf{DEq}}} = \big(dk_{(y_1,1)}^{\mathsf{ACE}_=}, \ldots, dk_{(y_\ell,\ell)}^{\mathsf{ACE}_=}\big)$ and a sanitized ciphertext $c' \coloneqq \big(c_1', \ldots, c_\ell'\big)$, the algorithm $\mathsf{ACE}_{\mathsf{DEq}}.\mathsf{Dec}$ computes for $i \in [\ell]$ the message $m_i \leftarrow \mathsf{ACE}_=.\mathsf{Dec}\big(dk_{(y_i,i)}^{\mathsf{ACE}_=}, c_i'\big)$. If $m_i \neq \bot$ for some $i \in [\ell]$, $\mathsf{ACE}_{\mathsf{DEq}}.\mathsf{Dec}$ outputs the first such m_i; otherwise it outputs $\bot$.

Modification detection: On input sanitizer parameters $sp^{\mathsf{ACE}_{\mathsf{DEq}}} \coloneqq \big(sp^{\mathsf{ACE}_=}, vk^{\mathsf{Sig}}, crs^{\mathsf{NIZK}}\big)$ and two ciphertexts $c = \big(c_1, \ldots, c_\ell, \pi^{\mathsf{NIZK}}\big)$ and $\tilde{c} \coloneqq \big(\tilde{c}_1, \ldots, \tilde{c}_\ell, \tilde{\pi}^{\mathsf{NIZK}}\big)$, the algorithm $\mathsf{ACE}_{\mathsf{DEq}}.\mathsf{DMod}$ checks for $i \in [\ell]$ whether $\mathsf{ACE}_=.\mathsf{DMod}\big(sp^{\mathsf{ACE}_=}, c_i, \tilde{c}_i\big) = 1$. If this is the case for some $i \in [\ell]$, it outputs 1; otherwise, it outputs 0.

Weak and strong anonymity. As we show below, our scheme enjoys weak anonymity. It is easy to see that it does not have strong anonymity: Given a decryption key for the role $(1, 2)$, one can decrypt ciphertexts encrypted under a key for the roles $(1, 1)$ and $(2, 2)$. One does, however, also learn which of the two components decrypted successfully. If it is the first one, the sender role must be $(1, 1)$, if it is the second one, the sender role must be $(2, 2)$. For similar reasons, we do not achieve strong sanitization security.

A similar construction can be used to achieve strong anonymity for less expressive policies: If a sender role still corresponds to a vector $(x_1, \ldots, x_\ell) \in \mathcal{D}^\ell$ but a receiver role only to one component $(j, y) \in [\ell] \times \mathcal{D}$, one can consider the policy that allows to receive if $x_j = y$. Now, we do not need several components for the decryption key and the problem sketched above disappears.

Proposition 6.4. *If $\mathsf{ACE}_=$ is correct and detectable, then the scheme $\mathsf{ACE}_{\mathsf{DEq}}$ from above is correct and detectable. If $\mathsf{ACE}_=$ is strongly detectable, then $\mathsf{ACE}_{\mathsf{DEq}}$ is also strongly detectable. More precisely, for all probabilistic algorithms $\mathcal{A}$, there exist probabilistic algorithms $\mathcal{A}_{\mathsf{corr}}$, $\mathcal{A}_{\mathsf{dtct}}$, $\mathcal{A}'_{\mathsf{dtct}}$, and $\mathcal{A}_{\mathsf{sdtct}}$ such that*

$$\begin{aligned} \mathsf{Adv}^{\mathsf{ACE\text{-}CORR}}_{\mathsf{ACE}_{\mathsf{DEq}},\mathcal{A}} &\le \mathsf{Adv}^{\mathsf{ACE\text{-}CORR}}_{\mathsf{ACE}_=,\mathcal{A}_{\mathsf{corr}}} + (\ell - 1) \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}DTCT}}_{\mathsf{ACE}_=,\mathcal{A}_{\mathsf{dtct}}},\\ \mathsf{Adv}^{\mathsf{ACE\text{-}DTCT}}_{\mathsf{ACE}_{\mathsf{DEq}},\mathcal{A}} &\le \ell \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}DTCT}}_{\mathsf{ACE}_=,\mathcal{A}'_{\mathsf{dtct}}},\\ \mathsf{Adv}^{\mathsf{ACE\text{-}sDTCT}}_{\mathsf{ACE}_{\mathsf{DEq}},\mathcal{A}} &\le \ell \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}sDTCT}}_{\mathsf{ACE}_=,\mathcal{A}_{\mathsf{sdtct}}}. \end{aligned}$$

Proof. We first prove correctness. Let $\mathcal{A}$ be a probabilistic algorithm and let $(m, \mathbf{x}, \mathbf{y})$ with $P_{\mathsf{DEq}}(\mathbf{x}, \mathbf{y}) = 1$ be the output of $\mathcal{A}$ in an execution of $\mathsf{Exp}^{\mathsf{ACE\text{-}CORR}}_{\mathsf{ACE}_{\mathsf{DEq}},\mathcal{A}}$. Correctness of the signature scheme and completeness of the NIZK imply that the verification in the sanitizer algorithm succeeds with probability 1. Note that $P_{\mathsf{DEq}}(\mathbf{x}, \mathbf{y}) = 1$ implies that there exists $i \in [\ell]$ with $x_i = y_i$. Let i_0 be the first such i. Then, $\mathcal{A}$ only wins the correctness game if either $\mathsf{ACE}_{=}.\mathsf{Dec}\big(dk^{\mathsf{ACE}_{=}}_{(y_{i_0},i_0)}, c'_{i_0}\big) \neq m$, or $\mathsf{ACE}_{=}.\mathsf{Dec}\big(dk^{\mathsf{ACE}_{=}}_{(y_i,i)}, c'_i\big) \neq \bot$ for some $i < i_0$. The probability of the former event is bounded by $\mathsf{Adv}^{\mathsf{ACE\text{-}CORR}}_{\mathsf{ACE}_{=},\mathcal{A}_{\mathsf{corr}}}$ where $\mathcal{A}_{\mathsf{corr}}$ emulates this experiment and returns $\big(m, (x_{i_0}, i_0), (y_{i_0}, i_0)\big)$. For the latter event, note that there are at most $\ell - 1$ such i, so the probability that $\mathsf{ACE}_{=}.\mathsf{Dec}$ returns a message different from $\bot$ for any of them can be bounded by $(\ell - 1) \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}DTCT}}_{\mathsf{ACE}_{\mathsf{DEq}},\mathcal{A}_{\mathsf{dtct}}}$ for the adversary $\mathcal{A}_{\mathsf{dtct}}$ that emulates the experiment and returns $\big(m, (x_i, i), (y_i, i)\big)$ for a uniformly chosen $i < i_0$.

For detectability, the adversary $\mathcal{A}'_{\mathsf{dtct}}$ emulates an execution of $\mathsf{Exp}^{\mathsf{ACE\text{-}DTCT}}_{\mathsf{ACE}_{\mathsf{DEq}},\mathcal{A}}$ and when $\mathcal{A}$ returns $(m, \mathbf{x}, \mathbf{y})$, $\mathcal{A}'_{\mathsf{dtct}}$ outputs $(m, (x_i, i), (y_i, i))$ for a uniformly chosen $i \in \{1, \ldots, \ell\}$. Note that $\mathcal{A}$ only wins if $P_{\mathsf{DEq}}(\mathbf{x}, \mathbf{y}) = 0$, which implies that $x_i \neq y_i$ for all $i \in [\ell]$. In this case, $\mathcal{A}$ wins if any of the ciphertext components decrypt to something different from $\bot$. Thus, $\mathcal{A}'_{\mathsf{dtct}}$ also wins if the component i was guesses correctly, which happens with probability $1/\ell$. The proof for strong detectability is similar, while $\mathcal{A}_{\mathsf{sdtct}}$ additionally outputs the randomness used for encrypting the chosen component when the randomness output by $\mathcal{A}$ is used to generate the whole ciphertext. □

The following theorem summarizes the security properties we prove for our scheme.

Theorem 6.5. *If F is pseudorandom, NIZK is zero-knowledge and extractable, Sig is EUF-CMA secure, and $\mathsf{ACE}_{=}$ is perfectly correct, strongly detectable, has NDTCT-FENC, and is PRV-CCA, wANON-CCA, SAN-CCA, RR, and UDEC secure, then the scheme $\mathsf{ACE}_{\mathsf{DEq}}$ from above has NDTCT-FENC and is PRV-CCA, wANON-CCA, SAN-CCA, RR, and UDEC secure.*

We prove this theorem in a sequence of lemmata proving the individual properties. We begin by showing that privacy and weak anonymity of the scheme follow from the corresponding properties of the underlying scheme for equality and the zero-knowledge property of the NIZK. Note that security of the PRF is not needed for this step since it is only used for the signatures, which are irrelevant here.

Lemma 6.6. *Let $\mathsf{ACE}_{\mathsf{DEq}}$, be the scheme from above, let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a probabilistic algorithm. Then, there exist probabilistic algorithms $\mathcal{A}_{\mathsf{ZK}}$, $\mathcal{A}_{\mathsf{ACE}}$, $\mathcal{A}'_{\mathsf{ZK}}$, and $\mathcal{A}'_{\mathsf{ACE}}$ (which are all roughly as efficient as emulating an execution of $\mathsf{Exp}^{\mathsf{ACE\text{-}PRV\text{-}ANON\text{-}CCA}}_{\mathsf{ACE}_{\mathsf{DEq}},\mathcal{A}}$) such that*

$$\mathsf{Adv}^{\mathsf{ACE\text{-}PRV\text{-}CCA}}_{\mathsf{ACE}_{\mathsf{DEq}},\mathcal{A}} \leq 2 \cdot \mathsf{Adv}^{\mathsf{NIZK\text{-}ZK}}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}}} + \ell \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}PRV\text{-}CCA}}_{\mathsf{ACE}_{=},\mathcal{A}_{\mathsf{ACE}}},$$
$$\mathsf{Adv}^{\mathsf{ACE\text{-}wANON\text{-}CCA}}_{\mathsf{ACE}_{\mathsf{DEq}},\mathcal{A}} \leq 2 \cdot \mathsf{Adv}^{\mathsf{NIZK\text{-}ZK}}_{\mathsf{NIZK},\mathcal{A}'_{\mathsf{ZK}}} + \ell \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}wANON\text{-}CCA}}_{\mathsf{ACE}_{=},\mathcal{A}'_{\mathsf{ACE}}}.$$

Proof. We only prove the statement about the privacy advantage. The proof for weak anonymity is completely analogous. We assume without loss of generality that $\mathcal{A}$ ensures $\mathbf{x}^0 = \mathbf{x}^1$ and $P(\mathbf{x}^0, \mathbf{y}) = 0$ for all $\mathbf{y} \in J$, since doing otherwise can only decrease the privacy advantage. Let $H_0 := \mathsf{Exp}^{\mathsf{ACE\text{-}PRV\text{-}ANON\text{-}CCA}}_{\mathsf{ACE}_{\mathsf{DEq}},\mathcal{A}}$ and let H_1 be as H_0 where we replace $crs^{\mathsf{NIZK}} \leftarrow \mathsf{NIZK.Gen}(1^\kappa)$ by $\big(crs^{\mathsf{NIZK}}, \tau^{\mathsf{NIZK}}\big) \leftarrow S_1^{\mathsf{NIZK}}(1^\kappa)$ in $\mathsf{ACE}_{\mathsf{DEq}}.\mathsf{Setup}$, and for the generation of the challenge ciphertext c^*, we replace $\pi^{\mathsf{NIZK}} \leftarrow \mathsf{NIZK.Prove}\big(crs^{\mathsf{NIZK}}, x, w\big)$ in $\mathsf{ACE}_{\mathsf{DEq}}.\mathsf{Enc}$ by

$\pi^{\mathsf{NIZK}} \leftarrow S_2^{\mathsf{NIZK}}(crs^{\mathsf{NIZK}}, \tau^{\mathsf{NIZK}}, x)$. It can be shown as in the proof of Lemma 6.3 that there exists a probabilistic algorithm $\mathcal{A}_{\mathsf{ZK}}$ such that

$$\Pr^{H_0}[b' = b] - \Pr^{H_1}[b' = b] = \mathsf{Adv}^{\mathsf{NIZK\text{-}ZK}}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}}}. \tag{4}$$

For $k \in \{0, \ldots, \ell\}$, we define $H_{2,k}$ as follows. It is identical to H_1 except that after $\mathcal{A}$ returns $(m_0, m_1, \mathbf{x}^0, \mathbf{x}^1, st)$, we replace the ciphertext components in c^* by

$$\begin{aligned} c_i &\leftarrow \mathsf{ACE}_{=}.\mathsf{Enc}\big(ek^{\mathsf{ACE}_=}_{(x_i^0,i)}, m_0; r_i\big) \quad (\text{for } 1 \le i \le k), \\ c_i &\leftarrow \mathsf{ACE}_{=}.\mathsf{Enc}\big(ek^{\mathsf{ACE}_=}_{(x_i^1,i)}, m_1; r_i\big) \quad (\text{for } k < i \le \ell). \end{aligned}$$

Note that $H_{2,0}$ corresponds to H_1 with $b = 1$ and $H_{2,\ell}$ corresponds to H_1 with $b = 0$. Now consider the adversary $\mathcal{A}_{\mathsf{ACE}}$ that on input sp chooses $k_0 \twoheadleftarrow \{1, \ldots, \ell\}$ uniformly at random and emulates an execution of H_1. It emulates the oracle $\mathcal{O}_G$ by obtaining all the required sub-keys from its own oracle $\mathcal{O}_G$. To emulate the oracle $\mathcal{O}_{SD}$, it first checks the NIZK proof as $\mathsf{ACE}_{\mathsf{DEq}}.\mathsf{San}$ and if the verification succeeds, it uses its oracle $\mathcal{O}_{SD}$ to sanitize and decrypt all ciphertext components. As $\mathsf{ACE}_{\mathsf{DEq}}.\mathsf{Dec}$, it outputs the first message different from $\perp$, or $\perp$ if no such message exists.

When $\mathcal{A}$ returns $(m_0, m_1, \mathbf{x}^0, \mathbf{x}^1, st)$, $\mathcal{A}_{\mathsf{ACE}}$ generates the challenge ciphertext c^* by encrypting m_0 under the key $ek^{\mathsf{ACE}_=}_{(x_i^0,i)}$ to obtain c_i for $1 \le i < k_0$, and by encrypting m_1 under the key $ek^{\mathsf{ACE}_=}_{(x_i^1,i)}$ for $k_0 < i \le \ell$, where these keys can obtain from $\mathcal{O}_G$ without changing the advantage. For the k_0-th component, it returns $\big(m_0, m_1, x^0_{k_0}, x^1_{k_0}\big)$ to the challenger and uses the obtained challenge ciphertext as c_{k_0}. It then proceeds with the emulation of H_1. It emulates the oracle $\mathcal{O}_G$ as above and the oracle $\mathcal{O}_{SD^*}$ as $\mathcal{O}_{SD}$ with the difference that if its own oracle returns `test` for any of the components, it returns `test` as well. Finally, when $\mathcal{A}_2$ returns b', $\mathcal{A}_{\mathsf{ACE}}$ returns $b'' \leftarrow b'$. Note that if $b = 0$ or $b = 1$, $\mathcal{A}_{\mathsf{ACE}}$ perfectly emulates an execution of H_{2,k_0} or H_{2,k_0-1}, respectively. Further note that since $\mathcal{A}$ by assumption does not query $\mathcal{O}_G$ on a decryption key for any $\mathbf{y}$ with $P(\mathbf{x}^0, \mathbf{y}) = 1$, $\mathcal{A}_{\mathsf{ACE}}$ also does not ask for a decryption that could decrypt the challenger ciphertext. Hence, $\mathcal{A}_{\mathsf{ACE}}$ wins if $b'' = b$ and we have

$$\begin{aligned} \mathsf{Adv}^{\mathsf{ACE\text{-}PRV\text{-}CCA}}_{\mathsf{ACE}_=,\mathcal{A}_{\mathsf{ACE}}} &= 2 \cdot \Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}PRV\text{-}ANON\text{-}CCA}}_{\mathsf{ACE}_=,\mathcal{A}_{\mathsf{ACE}}}}[b'' = b] - 1 \\ &= \Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}PRV\text{-}ANON\text{-}CCA}}_{\mathsf{ACE}_=,\mathcal{A}_{\mathsf{ACE}}}}[b'' = 1 \mid b = 1] - \Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}PRV\text{-}ANON\text{-}CCA}}_{\mathsf{ACE}_=,\mathcal{A}_{\mathsf{ACE}}}}[b'' = 1 \mid b = 0] \\ &= \sum_{k=1}^{\ell} \frac{1}{\ell} \Pr^{H_{2,k-1}}[b' = 1] - \sum_{k=1}^{\ell} \frac{1}{\ell} \Pr^{H_{2,k}}[b' = 1] \\ &= \big(\Pr^{H_{2,0}}[b' = 1] - \Pr^{H_{2,\ell}}[b' = 1]\big)/\ell \\ &= \big(\Pr^{H_1}[b' = 1 \mid b = 1] - \Pr^{H_1}[b' = 1 \mid b = 0]\big)/\ell. \end{aligned}$$

We therefore have that $2 \cdot \Pr^{H_1}[b' = b] - 1 = \ell \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}PRV\text{-}CCA}}_{\mathsf{ACE}_=,\mathcal{A}_{\mathsf{ACE}}}$. Combining this with equation (4) concludes the proof. □

Next, we sketch how to prove sanitization security.

Lemma 6.7. *If F is pseudorandom, NIZK is extractable, Sig is EUF-CMA secure, and $\mathsf{ACE}_=$ is perfectly correct, strongly detectable, and SAN-CCA secure, then the scheme $\mathsf{ACE}_{\mathsf{DEq}}$ from above is SAN-CCA secure.*

Proof sketch. The basic idea is to construct an adversary $\mathcal{A}_{\mathsf{SAN}}$ against the sanitization security of $\mathsf{ACE}_{=}$ that chooses $k_0 \twoheadleftarrow \{1,\dots\ell\}$ uniformly at random and emulates an execution of $\mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathsf{ACE}_{\mathsf{DEq}},\mathcal{A}}$. When $\mathcal{A}_1$ returns two ciphertexts c_0, c_1, $\mathcal{A}_{\mathsf{SAN}}$ gives the sanitized ciphertext $\big(c'_1,\dots,c'_\ell\big)$ to $\mathcal{A}_2$ where $c'_i \leftarrow \mathsf{ACE}_{=}.\mathsf{San}\big(c_{0,i}\big)$ for $1 \le i < k_0$, $c'_i \leftarrow \mathsf{ACE}_{=}.\mathsf{San}\big(c_{1,i}\big)$ for $k_0 < i \le \ell$, and c'_{k_0} is obtained from the challenger by submitting $\big(c_{0,k_0}, c_{1,k_0}\big)$. When $\mathcal{A}_2$ returns the bit b', $\mathcal{A}_{\mathsf{SAN}}$ returns the same bit b'. Note that $\mathcal{A}_{\mathsf{SAN}}$ wins if the bit is guessed correctly and if both returned ciphertexts sanitize properly and no decryption key has been obtained that decrypts any of the ciphertexts. If the last two conditions are not satisfied, then also $\mathcal{A}$ does not win. For the hybrid argument to go through, however, we need to ensure that $\mathcal{A}_{\mathsf{SAN}}$ still wins with probability $1/2$ when $\mathcal{A}$ violates one of these two conditions. To achieve this, $\mathcal{A}_{\mathsf{SAN}}$ needs to detect that this would next happen and in this case abort the emulation, return two valid ciphertexts (if not done already) and guess a uniform bit. To detect this event before it happens, extract witnesses from the ciphertexts returned by $\mathcal{A}_1$. If the ciphertexts are valid, the extractions are successful, the signature scheme is EUF-CMA secure, and the PRF is pseudorandom, then the ciphertexts have with overwhelming probability been obtained by encrypting messages with encryption keys that $\mathcal{A}_1$ has obtained from the oracle $\mathcal{O}_G$. Hence, $\mathcal{A}_{\mathsf{SAN}}$ knows in this case for which roles the messages have been encrypted. When $\mathcal{A}_2$ asks for a decryption key, $\mathcal{A}_{\mathsf{SAN}}$ checks whether the policy allows this key to decrypt any of the two ciphertexts. Given perfect correctness and strong detectability, the decryptions yield $\perp$ if and only if the policy does not allow decryption. Therefore, $\mathcal{A}_{\mathsf{SAN}}$ can detect when the bad event is about to happen and abort in this case. □

Non-detection of fresh encryptions directly follows from the same property of the underlying ACE scheme.

Lemma 6.8. *Let $\mathsf{ACE}_{\mathsf{DEq}}$, be the scheme from above and let $\mathcal{A}$ be an attacker on the non-detection of fresh encryptions. Then, there exists a probabilistic algorithm $\mathcal{A}'$ (which is roughly as efficient as emulating an execution of $\mathsf{Exp}^{\mathsf{ACE\text{-}NDTCT\text{-}FENC}}_{\mathsf{ACE}_{\mathsf{DEq}},\mathcal{A}}$) such that*

$$\mathsf{Adv}^{\mathsf{ACE\text{-}NDTCT\text{-}FENC}}_{\mathsf{ACE}_{\mathsf{DEq}},\mathcal{A}} \le \ell \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}NDTCT\text{-}FENC}}_{\mathsf{ACE}_{=},\mathcal{A}'}.$$

Proof. Let $\mathcal{A}'$ emulate an execution of $\mathsf{Exp}^{\mathsf{ACE\text{-}NDTCT\text{-}FENC}}_{\mathsf{ACE}_{\mathsf{DEq}},\mathcal{A}}$, using $\mathcal{O}_G$ to answer oracle queries from $\mathcal{A}$. When $\mathcal{A}$ returns $\big(m, \mathbf{x}, c = \big(c_1,\dots,c_\ell,\pi^{\mathsf{NIZK}}\big)\big)$, $\mathcal{A}'$ chooses $k \twoheadleftarrow \{1,\dots,\ell\}$ uniformly at random, and returns $\big(m,(x_k,k),c_k\big)$. If $\mathcal{A}$ wins, a fresh encryption of m under $\mathbf{x}$ is detected as a modification of c. Since encryption and modification detection are defined component-wise, this means that there exists a component k_0 such that a fresh encryption of m under (x_{k_0}, k_0) is detected to be a modification of c_{k_0}. Hence, $\mathcal{A}'$ also wins if additionally $k = k_0$, which happens with probability $1/\ell$. □

We finally prove role-respecting and uniform decryption security.

Lemma 6.9. *Let $\mathsf{ACE}_{\mathsf{DEq}}$, be the scheme from above and let $\mathcal{A}$ be a probabilistic algorithm that makes at most at most q_E queries to the oracle $\mathcal{O}_E$. Then, there exist probabilistic algorithms $\mathcal{A}_{\mathsf{PRF}}$, $\mathcal{A}_{\mathsf{ZK}_1}$, $\mathcal{A}_{\mathsf{ZK}_2}$, $\mathcal{A}_{\mathsf{Sig}}$, and $\mathcal{A}_{\mathsf{ACE}}$ (which are all roughly as efficient as emulating an execution of $\mathsf{Exp}^{\mathsf{ACE\text{-}URR}}_{\mathsf{ACE}_{\mathsf{DEq}},\mathcal{A}}$) such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{ACE\text{-}RR}}_{\mathsf{ACE}_{\mathsf{DEq}},\mathcal{A}} + \mathsf{Adv}^{\mathsf{ACE\text{-}UDEC}}_{\mathsf{ACE}_{\mathsf{DEq}},\mathcal{A}} &\le 2 \cdot \mathsf{Adv}^{\mathsf{PRF}}_{F,\mathcal{A}_{\mathsf{PRF}}} + 2 \cdot \mathsf{Adv}^{\mathsf{NIZK\text{-}ext}_1}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}_1}} + 2 \cdot \mathsf{Adv}^{\mathsf{NIZK\text{-}ext}_2}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}_2}} \\ &\quad + 2(q_E+1) \cdot \mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}}} + 2\ell \cdot \Big(\mathsf{Adv}^{\mathsf{ACE\text{-}RR}}_{\mathsf{ACE}_{=},\mathcal{A}_{\mathsf{ACE}}} + \mathsf{Adv}^{\mathsf{ACE\text{-}UDEC}}_{\mathsf{ACE}_{=},\mathcal{A}_{\mathsf{ACE}}}\Big).\end{aligned}$$

Proof sketch. As in the proof of Lemma D.5, we define hybrids $H_0 \coloneqq \mathsf{Exp}^{\mathsf{ACE\text{-}URR}}_{\mathsf{ACE}_{\mathsf{DEq}},\mathcal{A}}$, H_1 as H_0 where F_K is replaced by a uniform random function U, H_2 as H_1 where crs^{NIZK} is generated by E_1^{NIZK}, H_3 as H_2 where a witness $w = \left(ek^{\mathsf{ACE}_=}_{(x_1,1)}, \ldots, ek^{\mathsf{ACE}_=}_{(x_\ell,\ell)}, m, r_1, \ldots, r_\ell, vk^{\mathsf{Sig}}_{\mathbf{x}}, \sigma^{\mathsf{Sig}}_{\mathbf{x}}, \sigma^{\mathsf{Sig}}_{c}\right)$ is extracted from π^{NIZK} by E_2^{NIZK} after $\mathcal{A}$ returned $c \coloneqq \left(c_1, \ldots, c_\ell, \pi^{\mathsf{NIZK}}\right)$. We can bound the probability that no valid witness is extracted even though π^{NIZK} is a valid proof by the knowledge extraction advantage of a suitable adversary, and the probability that a valid witness was extracted and the contained encryption key was not obtained via an oracle call by the signature forgery advantage of another adversary as in the proof of Lemma D.5. If these events do not occur, the ciphertext c is an encryption of the message m under a valid key that was returned by $\mathcal{O}_G$. Hence, $\mathcal{A}$ can in this case only win the role-respecting game or the uniform decryption game if some ciphertext component violates one of these properties. We can construct an adversary $\mathcal{A}_{\mathsf{ACE}}$ that emulates the execution, guesses this component, and uses the corresponding ciphertext component to win the game for the underlying scheme for equality. □

7 Conclusion and Directions for Future Work

In this paper, we have critically revisited existing notions for access control encryption, proposed stronger security definitions, and presented a new scheme that provably achieves our strong requirements. The need for stronger notions is not only a theoretical one as we have shown: In particular, we have described a practical attack based on the observation that a semi-honest sanitizer might leak an unsanitized ciphertext to a dishonest party.

An important question is whether all realistic attacks are excluded by our definitions. Furthermore, we would like to understand the fundamental limits of ACE. This includes investigating in which scenarios it can or cannot be used. To settle these questions, the authors are currently working on a theoretical model to capture the use case of ACE in a simulation-based framework. Another interesting research direction is to find more efficient schemes for useful policies.

A Standard Cryptographic Primitives and Games

A.1 Decisional Diffie-Hellman Assumption

Definition A.1. Let $G = \langle g \rangle$ be a cyclic group of prime-order q and let g be a generator. Let $\mathcal{A}$ be a probabilistic algorithm that on input q, g, and three elements $X, Y, T \in G$ returns a bit d. Let $\mathsf{DDH}^{\mathrm{real}}_{g,\mathcal{A}}$ be the experiment where $\mathcal{A}$ is given two random group elements $X = g^a$, $Y = g^b$, and the value $T = g^{ab}$. Let $\mathsf{DDH}^{\mathrm{rand}}_{g,\mathcal{A}}$ be the experiment where $\mathcal{A}$ is given three random group elements $X = g^a$, $Y = g^b$, and $T = g^c$. We define the *decisional Diffie-Hellman (DDH) advantage* of $\mathcal{A}$ as

$$\mathsf{Adv}^{\mathsf{DDH}}_{g,\mathcal{A}} \coloneqq \Pr^{\mathsf{DDH}^{\mathrm{real}}_{g,\mathcal{A}}}[d = 1] - \Pr^{\mathsf{DDH}^{\mathrm{rand}}_{g,\mathcal{A}}}[d = 1].$$

The *decisional Diffie-Hellman (DDH) assumption* for the group G states that $\mathsf{Adv}^{\mathsf{DDH}}_{g,\mathcal{A}}$ is negligible for all efficient $\mathcal{A}$.

A.2 Pseudorandom Functions

Definition A.2. For $\kappa \in \mathbb{N}$, let $\mathcal{K}_\kappa$, $\mathcal{X}_\kappa$, and $\mathcal{Y}_\kappa$ be finite sets and let $F_\kappa \colon \mathcal{K}_\kappa \times \mathcal{X}_\kappa \to \mathcal{Y}_\kappa$ be a function. For $K \in \mathcal{K}_\kappa$, we use the notation $F_K \coloneqq F_\kappa(K, \cdot)$. Further let $\mathcal{A}$ be a probabilistic

algorithm and consider the experiment in which $\mathcal{A}$ outputs a bit after interacting with an oracle that either corresponds to F_K for a uniformly chosen $K \in \mathcal{K}_\kappa$, or to a uniformly chosen function $U\colon \mathcal{X}_\kappa \to \mathcal{Y}_\kappa$. We define the *pseudorandom function advantage* of $\mathcal{A}$ as

$$\mathsf{Adv}^{\mathsf{PRF}}_{F,\mathcal{A}} \coloneqq \Pr\big[\mathcal{A}^{F_K(\cdot)}(1^\kappa) = 1\big] - \Pr\big[\mathcal{A}^{U(\cdot)}(1^\kappa) = 1\big],$$

where the first probability is over the random coins of $\mathcal{A}$ and the choice of K, and the second probability is over the random coins of $\mathcal{A}$ and the choice of U. The function family F is called *pseudorandom* if $\mathsf{Adv}^{\mathsf{PRF}}_{F,\mathcal{A}}$ is negligible for all efficient $\mathcal{A}$.

A.3 Public-Key Encryption

Definition A.3. A *public-key encryption (PKE) scheme* consist of the following three PPT algorithms:

Key generation: The algorithm Gen on input a security parameter 1^κ, outputs a *public key* ek and a *private key* dk.

Encryption: The algorithm Enc on input a public key ek and a message $m \in \mathcal{M}$, outputs a *ciphertext* c.

Decryption: The algorithm Dec on input a private key dk and a ciphertext c, outputs a message $m \in \mathcal{M} \cup \{\perp\}$.

We require for all (ek, dk) in the range of Gen and all $m \in \mathcal{M}$ that

$$\mathsf{Dec}\big(dk, \mathsf{Enc}(ek, m)\big) = m$$

with probability 1.

Definition A.4. Let $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme and let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a pair of probabilistic algorithms. Consider the experiment $\mathsf{Exp}^{\mathsf{PKE\text{-}IND\text{-}CPA}}_{\mathcal{E},\mathcal{A}}$ in Figure 4. We define the *ciphertext indistinguishability under chosen-plaintext attacks advantage* of $\mathcal{A}$ as

$$\mathsf{Adv}^{\mathsf{PKE\text{-}IND\text{-}CPA}}_{\mathcal{E},\mathcal{A}} \coloneqq 2 \cdot \Pr\big[b' = b \ \wedge \ |m_0| = |m_1|\big] - 1$$

where the probability is over the randomness in $\mathsf{Exp}^{\mathsf{PKE\text{-}IND\text{-}CPA}}_{\mathcal{E},\mathcal{A}}$. The scheme $\mathcal{E}$ has *indistinguishable ciphertexts under chosen-plaintext attacks (is IND-CPA secure)* if $\mathsf{Adv}^{\mathsf{PKE\text{-}IND\text{-}CPA}}_{\mathcal{E},\mathcal{A}}$ is negligible for all efficient $\mathcal{A}$.

A.4 Digital Signature Schemes

Definition A.5. A *(digital) signature scheme* consist of the following three PPT algorithms:

Key generation: The algorithm Gen on input a security parameter 1^κ, outputs a *public key* vk and a *private key* sk.

Signing: The algorithm Sign on input a private key sk and a message $m \in \mathcal{M}$, outputs a *signature* σ.

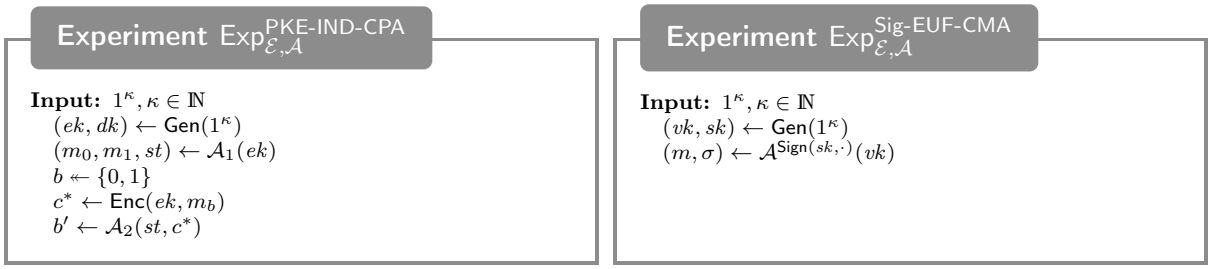

Figure 4: Experiments for the security definitions of public-key encryption and digital signature schemes.

Verification: The algorithm Ver is deterministic and on input a public key vk, a message m, and a signature σ, outputs a bit b (where $b = 1$ means "valid" and $b = 0$ means "invalid").

We require for all (vk, sk) in the range of Gen and all $m \in \mathcal{M}$ that

$$\mathsf{Ver}\big(vk, m, \mathsf{Sign}(sk, m)\big) = 1$$

with probability 1.

Definition A.6. Let $\mathcal{E} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ be a signature scheme and let $\mathcal{A}$ be a probabilistic algorithm. Consider the experiment $\mathsf{Exp}^{\text{Sig-EUF-CMA}}_{\mathcal{E},\mathcal{A}}$ in Figure 4 and let Q be the set of queries $\mathcal{A}$ issued to its oracle. We define the *existential unforgeability under adaptive chosen-message attacks advantage* of $\mathcal{A}$ as

$$\mathsf{Adv}^{\text{Sig-EUF-CMA}}_{\mathcal{E},\mathcal{A}} := \Pr\big[\mathsf{Ver}(vk, m, \sigma) = 1 \ \wedge \ m \notin Q\big],$$

where the probability is over the randomness in $\mathsf{Exp}^{\text{Sig-EUF-CMA}}_{\mathcal{E},\mathcal{A}}$. The scheme $\mathcal{E}$ is *existentially unforgeable under adaptive chosen-message attacks (EUF-CMA secure)* if $\mathsf{Adv}^{\text{Sig-EUF-CMA}}_{\mathcal{E},\mathcal{A}}$ is negligible for all efficient $\mathcal{A}$.

A.5 Non-Interactive Zero-Knowledge Proofs

We define non-interactive zero-knowledge proofs following Groth [Gro06].

Definition A.7. Let R be an efficiently computable binary relation and consider the *language* $L := \{x \mid \exists w \ (x, w) \in R\}$. A *non-interactive proof system* for L (or for R) consists of the following three PPT algorithms:

Key generation: The algorithm Gen on input a security parameter 1^κ, outputs a *common reference string* crs.

Proving: The algorithm Prove on input a common reference string crs, a *statement* x, and a *witness* w, outputs a *proof* π.

Verification: The algorithm Ver on input a common reference string crs, a statement x, and a proof π, outputs a bit b (where $b = 1$ means "accept" and $b = 0$ means "reject").

We require *perfect completeness*, i.e., for all crs in the range of Gen and for all $(x, w) \in R$, we have

$$\mathsf{Ver}\big(crs, x, \mathsf{Prove}(crs, x, w)\big) = 1$$

with probability 1.

Definition A.8 (Soundness)**.** Let $\mathcal{E} = (\mathsf{Gen}, \mathsf{Prove}, \mathsf{Ver})$ be a non-interactive proof system for a language L and let $\mathcal{A}$ be a probabilistic algorithm. We define the *soundness advantage* of $\mathcal{A}$ as

$$\mathsf{Adv}^{\mathsf{NIZK\text{-}snd}}_{\mathcal{E},\mathcal{A}} := \Pr^{crs \leftarrow \mathsf{Gen}(1^\kappa);\ (x,\pi) \leftarrow \mathcal{A}(crs)}\big[x \notin L \ \wedge\ \mathsf{Ver}(crs, x, \pi) = 1\big].$$

The scheme $\mathcal{E}$ is *computationally sound* if $\mathsf{Adv}^{\mathsf{NIZK\text{-}snd}}_{\mathcal{E},\mathcal{A}}$ is negligible for all efficient $\mathcal{A}$ and *perfectly sound* if $\mathsf{Adv}^{\mathsf{NIZK\text{-}snd}}_{\mathcal{E},\mathcal{A}} = 0$ for all $\mathcal{A}$.

Definition A.9 (Computational zero-knowledge)**.** Let $\mathcal{E} = (\mathsf{Gen}, \mathsf{Prove}, \mathsf{Ver})$ be a non-interactive proof system for a relation R and let $S = (S_1, S_2)$ be a pair of PPT algorithms, called *simulator*. Further let $S'(crs, \tau, x, w) = S_2(crs, \tau, x)$ for $(x, w) \in R$, and $S'(crs, \tau, x, w) = \texttt{failure}$ for $(x, w) \notin R$. We define the *zero-knowledge advantage* of a probabilistic algorithm $\mathcal{A}$ as

$$\mathsf{Adv}^{\mathsf{NIZK\text{-}ZK}}_{\mathcal{E},S,\mathcal{A}} := \Pr^{crs \leftarrow \mathsf{Gen}(1^\kappa)}\big[\mathcal{A}^{\mathsf{Prove}(crs,\cdot,\cdot)}(crs) = 1\big] - \Pr^{(crs,\tau) \leftarrow S_1(1^\kappa)}\big[\mathcal{A}^{S'(crs,\tau,\cdot,\cdot)}(crs) = 1\big].$$

We call $(\mathsf{Gen}, \mathsf{Prove}, \mathsf{Ver}, S_1, S_2)$ a *non-interactive zero-knowledge (NIZK) proof system* for R if $\mathsf{Adv}^{\mathsf{NIZK\text{-}ZK}}_{\mathcal{E},S,\mathcal{A}}$ is negligible for all efficient $\mathcal{A}$; it is called *single-theorem NIZK proof system* if $\mathsf{Adv}^{\mathsf{NIZK\text{-}ZK}}_{\mathcal{E},S,\mathcal{A}}$ is negligible for all efficient $\mathcal{A}$ that make at most one query to their oracle.

Definition A.10 (Knowledge extraction)**.** Let $\mathcal{E} = (\mathsf{Gen}, \mathsf{Prove}, \mathsf{Ver})$ be a non-interactive proof system for a relation R and let $E = (E_1, E_2)$ be a pair of PPT algorithms, called *knowledge extractor*. We define the *knowledge extraction advantages* of a probabilistic algorithm $\mathcal{A}$ as

$$\mathsf{Adv}^{\mathsf{NIZK\text{-}ext}_1}_{\mathcal{E},E,\mathcal{A}} := \Pr^{crs \leftarrow \mathsf{Gen}(1^\kappa)}\big[\mathcal{A}(crs) = 1\big] - \Pr^{(crs,\xi) \leftarrow E_1(1^\kappa)}\big[\mathcal{A}(crs) = 1\big],$$
$$\mathsf{Adv}^{\mathsf{NIZK\text{-}ext}_2}_{\mathcal{E},E,\mathcal{A}} := \Pr^{(crs,\xi) \leftarrow E_1(1^\kappa);\ (x,\pi) \leftarrow \mathcal{A}(crs);\ w \leftarrow E_2(crs,\xi,x,\pi)}\big[\mathsf{Ver}(crs, x, \pi) = 1 \ \wedge\ (x, w) \notin R\big].$$

We call $(\mathsf{Gen}, \mathsf{Prove}, \mathsf{Ver}, E_1, E_2)$ a *non-interactive proof of knowledge system* for R if $\mathsf{Adv}^{\mathsf{NIZK\text{-}ext}_1}_{\mathcal{E},E,\mathcal{A}}$ and $\mathsf{Adv}^{\mathsf{NIZK\text{-}ext}_2}_{\mathcal{E},E,\mathcal{A}}$ are negligible for all efficient $\mathcal{A}$.

Definition A.11 (Simulation soundness)**.** Let $\mathcal{E} = (\mathsf{Gen}, \mathsf{Prove}, \mathsf{Ver})$ be a non-interactive proof system for a language L, let $S = (S_1, S_2)$ be a pair of PPT algorithms, and let $\mathcal{A}$ be a probabilistic algorithm. Consider the experiment $\mathsf{Exp}^{\mathsf{NIZK\text{-}sim\text{-}snd}}_{\mathcal{E},S,\mathcal{A}}$ that executes $(crs, \tau) \leftarrow S_1(1^\kappa)$ and $(x, \pi) \leftarrow \mathcal{A}^{S_2(crs,\tau,\cdot)}(crs)$. Further let Q be the set of all (x', π') such that $\mathcal{A}$ queried x' to its oracle and received π' as a response. We define the *simulation soundness advantage* of $\mathcal{A}$ as

$$\mathsf{Adv}^{\mathsf{NIZK\text{-}sim\text{-}snd}}_{\mathcal{E},S,\mathcal{A}} := \Pr\big[(x, \pi) \notin Q \ \wedge\ x \notin L \ \wedge\ \mathsf{Ver}(crs, x, \pi) = 1\big].$$

We say $(\mathsf{Gen}, \mathsf{Prove}, \mathsf{Ver}, S_1, S_2)$ is *simulation sound* if $\mathsf{Adv}^{\mathsf{NIZK\text{-}sim\text{-}snd}}_{\mathcal{E},S,\mathcal{A}}$ is negligible for all efficient $\mathcal{A}$; it is *one-time simulation sound* if $\mathsf{Adv}^{\mathsf{NIZK\text{-}sim\text{-}snd}}_{\mathcal{E},S,\mathcal{A}}$ is negligible for all efficient $\mathcal{A}$ that make at most one query to the oracle S_2.

Note that in the above definition, $\mathcal{A}$ is allowed to issue queries $x' \notin L$ to its oracle. This means that soundness is preserved even if an adversary sees simulated proofs of false statements.

B Proof of Relation to the Original Security Notions

We prove the three claims in Theorem 4.7 as separate lemmata, starting with the payload-privacy no-read rule.

Lemma B.1. *Let* $\mathcal{E} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{San}, \mathsf{Dec}, \mathsf{DMod})$ *be an ACE with modification detection scheme and let* $\mathcal{E}' = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{San}, \mathsf{Dec})$ *be the corresponding ACE scheme. Further let* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *be a pair of probabilistic algorithms. Then, there exist adversaries* $\mathcal{A}_{\mathsf{PRV}}$ *and* $\mathcal{A}_{\mathsf{wANON}}$ *(both roughly as efficient as emulating an execution of* $\mathsf{Exp}^{\mathsf{ACE\text{-}no\text{-}read}}_{\mathcal{E},\mathcal{A}}$*) such that*

$$\mathsf{Adv}^{\mathsf{ACE\text{-}no\text{-}read,priv}}_{\mathcal{E}',\mathcal{A}} \leq \mathsf{Adv}^{\mathsf{ACE\text{-}PRV\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{PRV}}} + \mathsf{Adv}^{\mathsf{ACE\text{-}wANON\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{wANON}}}.$$

Proof. We assume without loss of generality that $\mathcal{A}$ ensures $|m_0| = |m_1|$ and $P(i_0, j) = P(i_1, j) = 0$ for all $j \in J$, where J is the set of all j such that $\mathcal{A}_1$ or $\mathcal{A}_2$ issued the query $(j, \texttt{rec})$ to the oracle $\mathcal{O}_G$. Let H be identical to $\mathsf{Adv}^{\mathsf{ACE\text{-}no\text{-}read,priv}}_{\mathcal{E}',\mathcal{A}}$ except that after $\mathcal{A}_1$ returns (m_0, m_1, i_0, i_1, st), i_1 is replaced by i_0. We first show that the probability that b is guessed correctly in H and $\mathsf{Adv}^{\mathsf{ACE\text{-}no\text{-}read,priv}}_{\mathcal{E}',\mathcal{A}}$ differ only negligibly if the scheme satisfies weak anonymity. Note that if $b = 0$, the two experiments are identical, which implies

$$\Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}no\text{-}read}}_{\mathcal{E}',\mathcal{A}}}\big[b' = b \mid b = 0\big] = \Pr^{H}\big[b' = b \mid b = 0\big]. \tag{5}$$

Claim 1. *There exists an adversary* $\mathcal{A}_{\mathsf{wANON}}$ *such that*

$$\Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}no\text{-}read}}_{\mathcal{E}',\mathcal{A}}}\big[b' = b \mid b = 1\big] - \Pr^{H}\big[b' = b \mid b = 1\big] = \mathsf{Adv}^{\mathsf{ACE\text{-}wANON\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{wANON}}}.$$

Proof of claim. We construct $\mathcal{A}_{\mathsf{wANON}}$ as follows. On input sp, it emulates an execution of $\mathsf{Exp}^{\mathsf{ACE\text{-}no\text{-}read}}_{\mathcal{E}',\mathcal{A}}$, where the oracles for $\mathcal{A}$ are emulated as follows.

$\mathcal{O}_G(\cdot,\cdot)$**:** Relay queries to the oracle $\mathcal{O}_G$ of $\mathsf{Exp}^{\mathsf{ACE\text{-}PRV\text{-}ANON\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{wANON}}}$.

$\mathcal{O}_E(\cdot,\cdot)$**:** On query (j, m), query $(j, \texttt{sen})$ to the oracle $\mathcal{O}_G$ to receive the encryption key ek_j. Then compute $c \leftarrow \mathsf{Enc}(ek_j, m)$ and return c.

When $\mathcal{A}$ outputs (m_0, m_1, i_0, i_1, st), $\mathcal{A}_{\mathsf{wANON}}$ gives (m_1, m_1, i_0, i_1) to the challenger to obtain a ciphertext c^*, which is given to $\mathcal{A}_2$. When $\mathcal{A}_2$ returns b', $\mathcal{A}_{\mathsf{wANON}}$ returns the same bit b'. Note that if $b = 0$, $\mathcal{A}_{\mathsf{wANON}}$ perfectly emulates H with $b = 1$, and if $b = 1$, $\mathcal{A}_{\mathsf{wANON}}$ perfectly emulates $\mathsf{Exp}^{\mathsf{ACE\text{-}no\text{-}read}}_{\mathcal{E}',\mathcal{A}}$ with $b = 1$. Hence,

$$\begin{aligned}
&\Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}no\text{-}read}}_{\mathcal{E}',\mathcal{A}}}\big[b' = b \mid b = 1\big] - \Pr^{H}\big[b' = b \mid b = 1\big] \\
&= \Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}PRV\text{-}ANON\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{wANON}}}}\big[b' = 1 \mid b = 1\big] - \Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}PRV\text{-}ANON\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{wANON}}}}\big[b' = 1 \mid b = 0\big] \\
&= 2 \cdot \left(\frac{1}{2}\Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}PRV\text{-}ANON\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{wANON}}}}\big[b' = b \mid b = 1\big] - \frac{1}{2}\Big(1 - \Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}PRV\text{-}ANON\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{wANON}}}}\big[b' = b \mid b = 0\big]\Big)\right) \\
&= 2 \cdot \Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}PRV\text{-}ANON\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{wANON}}}}\big[b' = b\big] - 1.
\end{aligned}$$

Note that $\mathcal{A}_{\mathsf{wANON}}$ returns the same message m_1 twice and we have $P(i_0, j) = P(i_1, j) = 0$ for all $j \in J$ by the assumption on $\mathcal{A}$. This implies $\mathsf{Adv}^{\mathsf{ACE\text{-}wANON\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{wANON}}} = 2 \cdot \Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}PRV\text{-}ANON\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{wANON}}}}\big[b' = b\big] - 1$ and concludes the proof of the claim. ◊

Combining Claim 1 and equation (5), we obtain

$$\begin{aligned}\Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}no\text{-}read}}_{\mathcal{E}',\mathcal{A}}}\big[b'=b\big] &= \frac{1}{2}\cdot\Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}no\text{-}read}}_{\mathcal{E}',\mathcal{A}}}\big[b'=b \mid b=0\big]+\frac{1}{2}\cdot\Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}no\text{-}read}}_{\mathcal{E}',\mathcal{A}}}\big[b'=b \mid b=1\big]\\ &= \frac{1}{2}\cdot\Pr^{H}\big[b'=b \mid b=0\big]+\frac{1}{2}\cdot\Big(\mathsf{Adv}^{\mathsf{ACE\text{-}wANON\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{wANON}}}+\Pr^{H}\big[b'=b \mid b=1\big]\Big)\\ &= \Pr^{H}\big[b'=b\big]+\frac{1}{2}\cdot\mathsf{Adv}^{\mathsf{ACE\text{-}wANON\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{wANON}}}.\end{aligned}$$

Hence,

$$\mathsf{Adv}^{\mathsf{ACE\text{-}no\text{-}read,priv}}_{\mathcal{E}',\mathcal{A}} = 2\cdot\Pr^{H}\big[b'=b\big]-1+\mathsf{Adv}^{\mathsf{ACE\text{-}wANON\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{wANON}}}. \tag{6}$$

We now construct the adversary $\mathcal{A}_{\mathsf{PRV}}$. When invoked on input sp, it starts an emulation of H by passing sp to $\mathcal{A}$. The oracles for $\mathcal{A}$ are emulated as in the proof of Claim 1. When $\mathcal{A}_1$ returns (m_0, m_1, i_0, i_1, st), $\mathcal{A}_{\mathsf{wANON}}$ gives (m_0, m_1, i_0, i_0) to the challenger to obtain a ciphertext c^*, which is then given to $\mathcal{A}_2$. When $\mathcal{A}_2$ returns b', $\mathcal{A}_{\mathsf{PRV}}$ returns the same bit b'. Note that the view of $\mathcal{A}$ in this emulation is identical to its view in H. Since $\mathcal{A}_{\mathsf{PRV}}$ returns the same role i_0 twice and $P(i_0, j) = 0$ for all $j \in J$ by the assumption on $\mathcal{A}$, we have

$$\mathsf{Adv}^{\mathsf{ACE\text{-}PRV\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{PRV}}} = 2\cdot\Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}PRV\text{-}ANON\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{PRV}}}}\big[b'=b\big]-1 = 2\cdot\Pr^{H}\big[b'=b\big]-1.$$

Using equation (6), we conclude

$$\mathsf{Adv}^{\mathsf{ACE\text{-}no\text{-}read,priv}}_{\mathcal{E}',\mathcal{A}} = \mathsf{Adv}^{\mathsf{ACE\text{-}PRV\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{PRV}}}+\mathsf{Adv}^{\mathsf{ACE\text{-}wANON\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{wANON}}}. \qquad \square$$

We next show that the sender-anonymity no-read rule is implied by strong sender anonymity.

Lemma B.2. *Let $\mathcal{E} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{San}, \mathsf{Dec}, \mathsf{DMod})$ be an ACE with modification detection scheme and let $\mathcal{E}' = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{San}, \mathsf{Dec})$ be the corresponding ACE scheme. Further let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a pair of probabilistic algorithms. Then, there exists an adversary $\mathcal{A}_{\mathsf{sANON}}$ (roughly as efficient as emulating an execution of $\mathsf{Exp}^{\mathsf{ACE\text{-}no\text{-}read}}_{\mathcal{E}',\mathcal{A}'}$) such that*

$$\mathsf{Adv}^{\mathsf{ACE\text{-}no\text{-}read,anon}}_{\mathcal{E}',\mathcal{A}} = \mathsf{Adv}^{\mathsf{ACE\text{-}sANON\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{sANON}}}.$$

Proof. We construct $\mathcal{A}_{\mathsf{sANON}}$ as follows. On input sp, it emulates an execution of $\mathsf{Exp}^{\mathsf{ACE\text{-}no\text{-}read}}_{\mathcal{E}',\mathcal{A}}$, where the oracles $\mathcal{O}_G$ and $\mathcal{O}_E$ for $\mathcal{A}$ are emulated as in the proof of Lemma B.1. When $\mathcal{A}_1$ returns (m_0, m_1, i_0, i_1, st), $\mathcal{A}_{\mathsf{sANON}}$ gives (m_0, m_1, i_0, i_1) to the challenger to obtain the ciphertext c^*. Then, $\mathcal{A}_2$ is invoked on input (st, c^*) and the oracles are emulated as before. When $\mathcal{A}_2$ terminates with output b', $\mathcal{A}_{\mathsf{sANON}}$ returns the same bit b'. We observe that the view $\mathcal{A}_{\mathsf{sANON}}$ emulates toward $\mathcal{A}$ is identical to the view of $\mathcal{A}$ in the experiment $\mathsf{Exp}^{\mathsf{ACE\text{-}no\text{-}read}}_{\mathcal{E}',\mathcal{A}}$. Thus,

$$\begin{aligned}&\mathsf{Adv}^{\mathsf{ACE\text{-}no\text{-}read,anon}}_{\mathcal{E}',\mathcal{A}}\\ &\quad= 2\cdot\Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}no\text{-}read}}_{\mathcal{E}',\mathcal{A}}}\big[b'=b \ \wedge\ m_0=m_1 \ \wedge\ \forall j\in J\ P(i_0,j)=P(i_1,j)\big]-1\\ &\quad= 2\cdot\Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}PRV\text{-}ANON\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{sANON}}}}\big[b'=b \ \wedge\ m_0=m_1 \ \wedge\ \forall j\in J\ P(i_0,j)=P(i_1,j)\big]-1\\ &\quad= \mathsf{Adv}^{\mathsf{ACE\text{-}sANON\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{sANON}}}.\end{aligned}$$

$\square$

To conclude the proof of Theorem 4.7, we prove the claim about the no-write with modification detection.

Lemma B.3. *Let $\mathcal{E} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{San}, \mathsf{Dec}, \mathsf{DMod})$ be an ACE with modification detection scheme and let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a pair of probabilistic algorithms. Then, there exist adversaries $\mathcal{A}_{\mathsf{SAN}}$, $\mathcal{A}_{\mathsf{RR}}$, and $\mathcal{A}_{\mathsf{CORR}}$ (all roughly as efficient as emulating an execution of $\mathsf{Exp}^{\mathsf{ACE\text{-}MD\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}}$) such that for policies P where for all i, one can efficiently find some j with $P(i,j) = 1$*

$$\mathsf{Adv}^{\mathsf{ACE\text{-}MD\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}} \leq \mathsf{Adv}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{SAN}}} + 4 \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}RR}}_{\mathcal{E},\mathcal{A}_{\mathsf{RR}}} + 2 \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}CORR}}_{\mathcal{E},\mathcal{A}_{\mathsf{CORR}}}.$$

Proof. We first construct the adversary $\mathcal{A}_{\mathsf{SAN}}$. When invoked on input sp, it gives sp to $\mathcal{A}_1$ and emulates $\mathsf{Exp}^{\mathsf{ACE\text{-}MD\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}}$. The oracles for $\mathcal{A}$ are emulated as follows.

$\mathcal{O}_G(\cdot,\cdot)$**:** Relay queries to the oracle $\mathcal{O}_G$ of $\mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{SAN}}}$.

$\mathcal{O}_{ES}(\cdot,\cdot)$**:** On query (j, m), $\mathcal{A}_{\mathsf{SAN}}$ queries $(j, \mathtt{sen})$ to its oracle $\mathcal{O}_G$ to receive the encryption key ek_j.[5] It then computes $c' \leftarrow \mathsf{San}(sp, \mathsf{Enc}(ek_j, m))$ and outputs c' to $\mathcal{A}$.

When $\mathcal{A}_1$ outputs (c_0, i', st), $\mathcal{A}_{\mathsf{SAN}}$ chooses a uniformly random message $m \twoheadleftarrow \mathcal{M}$, queries $(i', \mathtt{sen})$ to its oracle $\mathcal{O}_G$ to receive the encryption key $ek_{i'}$, and computes $c_1 \leftarrow \mathsf{Enc}(ek_{i'}, m)$. Then, $\mathcal{A}_{\mathsf{SAN}}$ gives (c_0, c_1) to the challenger to obtain a sanitized ciphertext c'_b. It then invokes $\mathcal{A}_2$ on input (st, c'_b) and emulates the oracles as above. When $\mathcal{A}_2$ outputs its guess b', $\mathcal{A}_{\mathsf{SAN}}$ outputs the same bit b' as its own guess. Note that the view $\mathcal{A}_{\mathsf{SAN}}$ emulates toward $\mathcal{A}$ is identical to the view of $\mathcal{A}$ in the experiment $\mathsf{Exp}^{\mathsf{ACE\text{-}MD\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}}$. Let W_{noW} and W_{san} be the events that $\mathcal{A}$ wins in the no-write with modification detection experiment and $\mathcal{A}_{\mathsf{SAN}}$ wins the sanitization experiment, respectively, i.e.,

$$\begin{aligned} W_{\mathrm{noW}} &:= \big[b' = b \ \wedge\ \mathtt{dct} = \mathtt{false} \ \wedge\ i' \in I_1 \\ &\qquad\qquad \wedge\ \forall i \in I_1\ \forall j \in J\ P(i,j) = 0 \ \wedge\ \mathsf{San}(sp, c_0) \neq \bot\big], \\ W_{\mathrm{san}} &:= \big[b' = b \ \wedge\ c'_0 \neq \bot \neq c'_1 \ \wedge\ \forall j \in J\ m_{0,j} = m_{1,j} = \bot\big]. \end{aligned}$$

Further consider the events

$$\begin{aligned} C &:= [\mathsf{San}(sp, c_1) \neq \bot], \\ R &:= [\forall j \in J\ \mathsf{Dec}\big(\mathsf{Gen}(msk, j, \mathtt{rec}), \mathsf{San}(sp, c_0)\big) = \mathsf{Dec}\big(\mathsf{Gen}(msk, j, \mathtt{rec}), \mathsf{San}(sp, c_1)\big) = \bot]. \end{aligned}$$

We then have

$$\Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{SAN}}}}[W_{\mathrm{san}}] \geq \Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}MD\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}}}[W_{\mathrm{noW}} \cap C \cap R]. \tag{7}$$

We next show that the events $\neg C$ and $\neg R$ only occur with negligible probability if the ACE scheme is correct and role-respecting, respectively.

Claim 1. *There exists an adversary $\mathcal{A}_{\mathsf{CORR}}$ (roughly as efficient as emulating an execution of $\mathsf{Exp}^{\mathsf{ACE\text{-}MD\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}}$) such that*

$$\Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}MD\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}}}[\neg C] \leq \mathsf{Adv}^{\mathsf{ACE\text{-}CORR}}_{\mathcal{E},\mathcal{A}_{\mathsf{CORR}}}.$$

Proof of claim. On input sp, the adversary $\mathcal{A}_{\mathsf{CORR}}$ begins an emulation of $\mathsf{Exp}^{\mathsf{ACE\text{-}MD\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}}$ as $\mathcal{A}_{\mathsf{SAN}}$ above. When $\mathcal{A}_1$ outputs (c_0, i', st), $\mathcal{A}_{\mathsf{CORR}}$ chooses a uniformly random message $m \twoheadleftarrow \mathcal{M}$ and finds j with $P(i', j) = 1$. It finally returns (m, i', j). By definition of Dec, we have $\mathsf{Dec}(\mathsf{Gen}(msk, j, \mathtt{rec}), \bot) = \bot$. Hence, if $\neg C$ occurs, then encrypting m for role i' and sanitizing and decrypting the result yields $\bot \neq m$. Therefore, $\mathcal{A}_{\mathsf{CORR}}$ wins the correctness game in this case, which implies the claim. ◊

[5] Looking ahead, we note that obtaining additional encryption keys is not problematic in the sanitization game, since the winning condition does not restrict the obtained encryption keys.

Claim 2. *There exists an adversary $\mathcal{A}_{\mathsf{RR}}$ (roughly as efficient as emulating an execution of $\mathsf{Exp}^{\mathsf{ACE\text{-}MD\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}}$) such that*

$$\Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}MD\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}}}[W_{\mathrm{noW}} \cap C \cap \neg R] \leq 2 \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}RR}}_{\mathcal{E},\mathcal{A}_{\mathsf{RR}}}.$$

Proof of claim. When invoked on input sp, $\mathcal{A}_{\mathsf{RR}}$ internally emulates an execution of $\mathcal{A}$ on input sp and emulates the oracles as follows.

$\mathcal{O}_G(\cdot,\cdot)$**:** Relay queries to the oracle $\mathcal{O}_G$ of $\mathsf{Exp}^{\mathsf{ACE\text{-}URR}}_{\mathcal{E},\mathcal{A}_{\mathsf{RR}}}$.

$\mathcal{O}_{ES}(\cdot,\cdot)$**:** On query (j,m), query (j,m) to $\mathcal{O}_E$ to receive the ciphertext c. Then, compute $c' \leftarrow \mathsf{San}(sp,c)$ and return c'.

When $\mathcal{A}_1$ outputs (c_0,i',st), $\mathcal{A}_{\mathsf{RR}}$ chooses a uniformly random message $m \twoheadleftarrow \mathcal{M}$, queries $(i',\mathtt{sen})$ to its oracle $\mathcal{O}_G$ to receive the encryption key $ek_{i'}$, and computes $c_1 \leftarrow \mathsf{Enc}(ek_{i'},m)$. Then, $\mathcal{A}_{\mathsf{RR}}$ chooses $c \twoheadleftarrow \{c_0,c_1\}$ uniformly at random and outputs c to the challenger. Let W_{noW} be the event that $\mathcal{A}_{\mathsf{RR}}$ wins the role-respecting game, i.e.,

$$W_{\mathrm{RR}} \coloneqq \big[c' \neq \bot \ \wedge\ \mathtt{dct} = \mathtt{false} \ \wedge\ \neg\big(\exists i \in I\ \forall j \in J\ (m_j \neq \bot \leftrightarrow P(i,j) = 1)\big)\big]$$

Note that W_{noW} and C imply that $c' \neq \bot$, $\mathtt{dct} = \mathtt{false}$, and $\forall i \in I\ \forall j \in J\ P(i,j) = 0$. Hence, if we additionally have $\neg R$, at least one of the two possible choices for c yield $m_j \neq \bot$ for some $j \in J$, and thus

$$\Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}MD\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}}}[W_{\mathrm{noW}} \cap C \cap \neg R] \leq 2 \cdot \Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}URR}}_{\mathcal{E},\mathcal{A}_{\mathsf{RR}}}}[W_{\mathrm{RR}}] = 2 \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}RR}}_{\mathcal{E},\mathcal{A}_{\mathsf{RR}}}. \qquad \lozenge$$

Combining equation (7) and Claims 1 and 2, we obtain

$$\begin{aligned}\Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}MD\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}}}[W_{\mathrm{noW}}] &\leq \Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}MD\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}}}[W_{\mathrm{noW}} \cap C \cap R] \\ &\quad + \Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}MD\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}}}[W_{\mathrm{noW}} \cap C \cap \neg R] + \Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}MD\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}}}[\neg C] \\ &\leq \Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{SAN}}}}[W_{\mathrm{san}}] + 2 \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}RR}}_{\mathcal{E},\mathcal{A}_{\mathsf{RR}}} + \mathsf{Adv}^{\mathsf{ACE\text{-}CORR}}_{\mathcal{E},\mathcal{A}_{\mathsf{CORR}}}.\end{aligned}$$

We can thus conclude

$$\begin{aligned}\mathsf{Adv}^{\mathsf{ACE\text{-}MD\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}} &= 2 \cdot \Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}MD\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}}}[W_{\mathrm{noW}}] - 1 \\ &\leq \underbrace{2 \cdot \Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{SAN}}}}[W_{\mathrm{san}}] - 1}_{= \mathsf{Adv}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{SAN}}}} + 4 \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}RR}}_{\mathcal{E},\mathcal{A}_{\mathsf{RR}}} + 2 \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}CORR}}_{\mathcal{E},\mathcal{A}_{\mathsf{CORR}}}. \qquad \square\end{aligned}$$

We finally prove Theorem 4.8, which we first restate.

Theorem 4.8. *Let $\mathcal{E} = (\mathsf{Setup},\mathsf{Gen},\mathsf{Enc},\mathsf{San},\mathsf{Dec},\mathsf{DMod})$ be an ACE with modification detection scheme such that $\Pr[\mathsf{DMod}(sp,c_0,c_1)=1] = \Pr[\mathsf{DMod}(sp,c_1,c_0)=1]$ for all sp returned by Setup and all ciphertexts $c_0,c_1 \in \mathcal{C}$. Further let $\mathcal{E}' = (\mathsf{Setup},\mathsf{Gen},\mathsf{Enc},\mathsf{San},\mathsf{Dec})$ be the corresponding ACE scheme. If $\mathcal{E}$ is correct, detectable, has NDTCT-FENC, and is sSAN-CCA and RR secure, then $\mathcal{E}'$ satisfies the no-write rule for policies P such that for all i, one can efficiently find some j with $P(i,j)=1$. More precisely, for all adversaries $\mathcal{A}$ that make at most q_{ES} queries to the oracle $\mathcal{O}_{ES}$ and at most q_{dk} queries of the form $(\cdot,\mathtt{rec})$ to $\mathcal{O}_G$, there exist adversaries $\mathcal{A}_{\mathsf{SAN}}$,*

$\mathcal{A}_{\mathsf{RR}}$, $\mathcal{A}_{\mathsf{sSAN}}$, $\mathcal{A}_{\mathsf{NDTCT}}$, $\mathcal{A}_{\mathsf{CORR}}$, *and* $\mathcal{A}_{\mathsf{dtct}}$ *(all roughly as efficient as emulating an execution of* $\mathsf{Exp}^{\mathsf{ACE\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}}$*) such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{ACE\text{-}no\text{-}write}}_{\mathcal{E}',\mathcal{A}} &\le \mathsf{Adv}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{SAN}}} + 4\cdot\mathsf{Adv}^{\mathsf{ACE\text{-}RR}}_{\mathcal{E},\mathcal{A}_{\mathsf{RR}}} + 2q_{ES}\cdot\mathsf{Adv}^{\mathsf{ACE\text{-}sSAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{sSAN}}}\\ &\quad + 4q_{ES}\cdot\mathsf{Adv}^{\mathsf{ACE\text{-}NDTCT\text{-}FENC}}_{\mathcal{E},\mathcal{A}_{\mathsf{NDTCT}}} + (8q_{ES}q_{dk}+2)\cdot\mathsf{Adv}^{\mathsf{ACE\text{-}CORR}}_{\mathcal{E},\mathcal{A}_{\mathsf{CORR}}} + 8q_{ES}q_{dk}\cdot\mathsf{Adv}^{\mathsf{ACE\text{-}DTCT}}_{\mathcal{E},\mathcal{A}_{\mathsf{dtct}}}.\end{aligned}$$

Proof. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an adversary and consider $\mathsf{Exp}^{\mathsf{ACE\text{-}MD\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}}$. Let W_{noW} and $W_{\text{MD-noW}}$ be the events that $\mathcal{A}$ wins the no-write and no-write with modification detection game, respectively. Note that we have $W_{\text{MD-noW}} = W_{\text{noW}} \cap [\texttt{dct} = \texttt{false}]$. Hence,

$$\begin{aligned}\mathsf{Adv}^{\mathsf{ACE\text{-}no\text{-}write}}_{\mathcal{E}',\mathcal{A}} &= 2\cdot\Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}MD\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}}}[W_{\text{noW}}] - 1\\ &= 2\Big(\Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}MD\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}}}[W_{\text{MD-noW}}] + \Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}MD\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}}}\big[W_{\text{noW}}\cap[\texttt{dct} = \texttt{true}]\big]\Big) - 1\\ &\le \mathsf{Adv}^{\mathsf{ACE\text{-}MD\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}} + 2\cdot\Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}MD\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}}}[\texttt{dct} = \texttt{true}].\end{aligned}$$

Lemma B.3 implies that there exist adversaries $\mathcal{A}_{\mathsf{SAN}}$, $\mathcal{A}_{\mathsf{RR}}$, and $\mathcal{A}'_{\mathsf{CORR}}$ (all roughly as efficient as emulating an execution of $\mathsf{Exp}^{\mathsf{ACE\text{-}MD\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}}$) such that

$$\begin{aligned}\mathsf{Adv}^{\mathsf{ACE\text{-}no\text{-}write}}_{\mathcal{E}',\mathcal{A}} &\le \mathsf{Adv}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{SAN}}} + 4\cdot\mathsf{Adv}^{\mathsf{ACE\text{-}RR}}_{\mathcal{E},\mathcal{A}_{\mathsf{RR}}} + 2\cdot\mathsf{Adv}^{\mathsf{ACE\text{-}CORR}}_{\mathcal{E},\mathcal{A}'_{\mathsf{CORR}}}\\ &\qquad + 2\cdot\Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}MD\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}}}[\texttt{dct} = \texttt{true}]. \quad (8)\end{aligned}$$

To bound the probability of $[\texttt{dct} = \texttt{true}]$, we construct the adversary $\mathcal{A}_{\mathsf{sSAN}}$. When invoked on input sp, it first chooses $q_0 \twoheadleftarrow \{1,\ldots,q_{ES}\}$ uniformly at random, sets $k \leftarrow 1$ and internally emulates an execution of $\mathcal{A}$ on input sp. Oracle queries by $\mathcal{A}$ are answered as follows:

$\mathcal{O}_G(\cdot,\cdot)$**:** Relay queries to the oracle $\mathcal{O}_G$ of $\mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{sSAN}}}$.

$\mathcal{O}_{ES}(\cdot,\cdot)$**:** On query (i,m), if $k \neq q_0$, $\mathcal{A}_{\mathsf{sSAN}}$ queries $(i,\mathtt{sen})$ to its oracle $\mathcal{O}_G$ to receive the encryption key ek_i. It then computes $c' \leftarrow \mathsf{San}(sp, \mathsf{Enc}(ek_i, m))$ and outputs c' to $\mathcal{A}$. Finally, it sets $k \leftarrow k+1$.

If $k = q_0$, then $\mathcal{A}_{\mathsf{sSAN}}$ queries $(i,\mathtt{sen})$ to its oracle $\mathcal{O}_G$ to receive the encryption key ek_i. It then creates two independent encryptions of m by computing $\tilde{c}_0 \leftarrow \mathsf{Enc}(ek_i, m)$ and $\tilde{c}_1 \leftarrow \mathsf{Enc}(ek_i, m)$, sets $i_{q_0} \leftarrow i$, $m_{q_0} \leftarrow m$, $k \leftarrow k+1$, and gives $\tilde{c}_0, \tilde{c}_1$ to the challenger to obtain $\tilde{c}'_b$.

If $\mathcal{A}_1$ terminates before $k = q_0$ is reached, $\mathcal{A}_{\mathsf{sSAN}}$ gives two fresh encryptions of some fixed message m_{q_0} for a fixed role i_{q_0} to the challenger and then returns a uniform bit $b' \twoheadleftarrow \{0,1\}$. Otherwise, when $\mathcal{A}_1$ returns i' and c_0, $\mathcal{A}_{\mathsf{sSAN}}$ evaluates $d_0 \leftarrow \mathsf{DMod}(sp, \tilde{c}_0, c_0)$ and $d_1 \leftarrow \mathsf{DMod}(sp, \tilde{c}_1, c_0)$. If $d_0 = d_1$, then $\mathcal{A}_{\mathsf{sSAN}}$ also returns a uniform bit; if $d_{b'} = 1$ for exactly one $b' \in \{0,1\}$, $\mathcal{A}_{\mathsf{sSAN}}$ returns b'.

Let Q be the event that $d_0 = 1$ or $d_1 = 1$ and let D be the event that $d_{1-b} = 1$. Note that if Q and $\neg D$ occur, $\mathcal{A}_{\mathsf{sSAN}}$ returns the correct bit $b' = b$. Moreover, if Q does not occur, $\mathcal{A}_{\mathsf{sSAN}}$ returns a uniform bit. Hence,

$$\begin{aligned}\Pr[b' = b] &= \Pr\big[[b' = b]\cap Q\cap\neg D\big] + \Pr\big[[b' = b]\cap\neg(Q\cap\neg D)\big]\\ &\ge \Pr[Q\cap\neg D] + \Pr\big[[b' = b]\cap\neg Q\big]\\ &= \Pr[Q\cap\neg D] + \Pr[b' = b \mid \neg Q]\cdot\Pr[\neg Q]\\ &= \Pr[Q\cap\neg D] + \tfrac{1}{2}\cdot(1 - \Pr[Q]),\end{aligned}$$

where all probabilities are in $\mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{sSAN}}}$. This implies

$$\Pr[Q] = \Pr[Q \cap D] + \Pr[Q \cap \neg D] \leq \Pr[D] + \Pr[b' = b] - \tfrac{1}{2} + \tfrac{1}{2}\Pr[Q]$$

and therefore

$$\Pr[Q] \leq 2 \cdot \Pr[D] + 2 \cdot \Pr[b' = b] - 1.$$

Note that if $[\mathtt{dct} = \mathtt{true}]$ occurs in $\mathsf{Exp}^{\mathsf{ACE\text{-}MD\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}}$, then Q occurs in $\mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{sSAN}}}$ with probability at least $1/q_{ES}$. We can therefore conclude that

$$\begin{aligned}\Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}MD\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}}}[\mathtt{dct} = \mathtt{true}] &\leq q_{ES} \cdot \Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{sSAN}}}}[Q] \\ &\leq 2q_{ES} \cdot \Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{sSAN}}}}[D] + q_{ES}\Big(2 \cdot \Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{sSAN}}}}[b' = b] - 1\Big).\end{aligned}$$

Let W_{sSAN} be the event that $\mathcal{A}_{\mathsf{sSAN}}$ wins the strong sanitization game and consider the events

$$\begin{aligned}C &:= \big[\tilde{c}_0' \neq \bot \neq \tilde{c}_1' \ \wedge\ \forall j \in J\ (P(i_{q_0}, j) = 1 \rightarrow m_{0,j} = m_{1,j} = m_{q_0})\big], \\ R &:= \big[\forall j \in J\ (P(i_{q_0}, j) = 0 \rightarrow m_{0,j} = m_{1,j} = \bot)\big].\end{aligned}$$

We then have that $[b' = b]$, C, and R together imply W_{sSAN}. Thus,

$$\begin{aligned}\Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{sSAN}}}}[b' = b] &= \Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{sSAN}}}}\big[[b' = b] \cap C \cap R\big] + \Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{sSAN}}}}\big[[b' = b] \cap \neg(C \cap R)\big] \\ &\leq \Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{sSAN}}}}[W_{\text{sSAN}}] + \Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{sSAN}}}}[\neg C \cup \neg R].\end{aligned}$$

Together with the previous result, this yields

$$\begin{aligned}&\Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}MD\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}}}[\mathtt{dct} = \mathtt{true}] \\ &\leq 2q_{ES} \cdot \Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{sSAN}}}}[D] + q_{ES}\Big[2 \cdot \Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{sSAN}}}}[W_{\text{sSAN}}] - 1 + 2 \cdot \Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{sSAN}}}}[\neg C \cup \neg R]\Big] \\ &= 2q_{ES} \cdot \Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{sSAN}}}}[D] + q_{ES} \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}sSAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{sSAN}}} + 2q_{ES} \cdot \Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{sSAN}}}}[\neg C \cup \neg R].\end{aligned}$$

Now consider $\mathcal{A}_{\mathsf{NDTCT}}$ that emulates $\mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{sSAN}}}$ and outputs (m_{q_0}, i_{q_0}, c_0). Note that the view of $\mathcal{A}_1$ in the emulation is independent of $\tilde{c}_{1-b}$. One can therefore assume that $\tilde{c}_{1-b}$ is generated after $\mathcal{A}_1$ outputs c_0, as c^* in $\mathsf{Exp}^{\mathsf{ACE\text{-}NDTCT\text{-}FENC}}_{\mathcal{E},\mathcal{A}_{\mathsf{NDTCT}}}$. By assumption, we have $\Pr[\mathsf{DMod}(sp, \tilde{c}_{1-b}, c_0) = 1] = \Pr[\mathsf{DMod}(sp, c_0, \tilde{c}_{1-b}) = 1]$, and therefore

$$\Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{sSAN}}}}[D] = \mathsf{Adv}^{\mathsf{ACE\text{-}NDTCT\text{-}FENC}}_{\mathcal{E},\mathcal{A}_{\mathsf{NDTCT}}}.$$

Further consider $\mathcal{A}''_{\mathsf{CORR}}$ that emulates $\mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{sSAN}}}$ and if there exists $j \in J$ with $P(i_{q_0}, j) = 1$, it chooses $j \twoheadleftarrow J$ uniformly at random and outputs (m_{q_0}, i_{q_0}, j). If such $j \in J$ does not exist, $\mathcal{A}''_{\mathsf{CORR}}$ finds $j \in [n]$ with $P(i_{q_0}, j) = 1$ and then outputs (m_{q_0}, i_{q_0}, j). Note that $m_{0,j} = m_{1,j} = m_{q_0}$ implies $\tilde{c}_0' \neq \bot \neq \tilde{c}_1'$ since $\bot$ decrypts to $\bot$. Hence, if such $j \in J$ exists and $\neg C$ occurs, $\mathcal{A}''_{\mathsf{CORR}}$ wins the correctness game with probability at least $1/(2|J|) \geq 1/(2q_{dk})$; and if no such $j \in J$ exists and $\neg C$ occurs, $\mathcal{A}''_{\mathsf{CORR}}$ wins with probability at least $1/2$. The factor $1/2$ is due to the fact that the message is encrypted twice in $\mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{sSAN}}}$ but only once in the correctness experiment. Overall, we get

$$\Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{sSAN}}}}[\neg C] \leq 2q_{dk} \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}CORR}}_{\mathcal{E},\mathcal{A}''_{\mathsf{CORR}}}.$$

Finally consider $\mathcal{A}_{\mathsf{dtct}}$ that emulates $\mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{sSAN}}}$, chooses $j \leftarrow\!\!\!\!\leftarrow J$ uniformly at random and outputs (m_{q_0}, i_{q_0}, j). If $\neg R$ occurs, $\mathcal{A}_{\mathsf{dtct}}$ wins the detectability game with probability at least $1/(2q_{dk})$. Hence,

$$\Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{sSAN}}}}[\neg R] \leq 2q_{dk} \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}DTCT}}_{\mathcal{E},\mathcal{A}_{\mathsf{dtct}}}.$$

Combining our results, we obtain

$$\begin{aligned}\Pr^{\mathsf{Exp}^{\mathsf{ACE\text{-}MD\text{-}no\text{-}write}}_{\mathcal{E},\mathcal{A}}}[\mathtt{dct} = \mathtt{true}] \leq 2q_{ES} \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}NDTCT\text{-}FENC}}_{\mathcal{E},\mathcal{A}_{\mathsf{NDTCT}}} + q_{ES} \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}sSAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{sSAN}}} \\ + 4q_{ES}q_{dk} \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}CORR}}_{\mathcal{E},\mathcal{A}''_{\mathsf{CORR}}} + 4q_{ES}q_{dk} \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}DTCT}}_{\mathcal{E},\mathcal{A}_{\mathsf{dtct}}}.\end{aligned}$$

Together with equation (8), this yields

$$\begin{aligned}\mathsf{Adv}^{\mathsf{ACE\text{-}no\text{-}write}}_{\mathcal{E}',\mathcal{A}} \leq \mathsf{Adv}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{SAN}}} + 4 \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}RR}}_{\mathcal{E},\mathcal{A}_{\mathsf{RR}}} + 2 \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}CORR}}_{\mathcal{E},\mathcal{A}'_{\mathsf{CORR}}} + 4q_{ES} \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}NDTCT\text{-}FENC}}_{\mathcal{E},\mathcal{A}_{\mathsf{NDTCT}}} \\ + 2q_{ES} \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}sSAN\text{-}CCA}}_{\mathcal{E},\mathcal{A}_{\mathsf{sSAN}}} + 8q_{ES}q_{dk} \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}CORR}}_{\mathcal{E},\mathcal{A}''_{\mathsf{CORR}}} + 8q_{ES}q_{dk} \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}DTCT}}_{\mathcal{E},\mathcal{A}_{\mathsf{dtct}}}.\end{aligned}$$

For the adversary $\mathcal{A}_{\mathsf{CORR}}$ that runs $\mathcal{A}'_{\mathsf{CORR}}$ with probability $\frac{2}{8q_{ES}q_{dk}+2}$ and $\mathcal{A}''_{\mathsf{CORR}}$ with probability $\frac{8q_{ES}q_{dk}}{8q_{ES}q_{dk}+2}$, we have $(8q_{ES}q_{dk}+2) \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}CORR}}_{\mathcal{E},\mathcal{A}_{\mathsf{CORR}}} = 2 \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}CORR}}_{\mathcal{E},\mathcal{A}'_{\mathsf{CORR}}} + 8q_{ES}q_{dk} \cdot \mathsf{Adv}^{\mathsf{ACE\text{-}CORR}}_{\mathcal{E},\mathcal{A}''_{\mathsf{CORR}}}$ and the claim of the theorem follows. □

C Proofs of Privacy and Anonymity of the sPKE Scheme

To complete the proof of Theorem 5.9, we first show that our sPKE scheme is IND-CCA secure. The proof follows Lindell's proof for the construction of an IND-CCA secure public-key encryption scheme from a IND-CPA secure one [Lin06].

Lemma C.1. *Let* sPKE *be the scheme from Section 5.2 and let* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *be a pair of probabilistic algorithms such that* $\mathcal{A}_1$ *and* $\mathcal{A}_2$ *together make at most* q_G *queries to* $\mathcal{O}_G$ *and at most* q_{SD} *queries to* $\mathcal{O}_{SD}$*. Then, there exist adversaries* $\mathcal{A}_{\mathsf{DDH}}$*,* $\mathcal{A}_{\mathsf{ZK}}$*,* $\mathcal{A}_{\mathsf{snd}}$*, and* $\mathcal{A}_{\mathsf{Sig}}$ *(which are all roughly as efficient as emulating an execution of* $\mathsf{Exp}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}}$*) such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}} \leq 4 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{g,\mathcal{A}_{\mathsf{DDH}}} + 2 \cdot \mathsf{Adv}^{\mathsf{NIZK\text{-}ZK}}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}}} + 2q_{SD} \cdot \mathsf{Adv}^{\mathsf{NIZK\text{-}sim\text{-}snd}}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{snd}}} \\ + 2 \cdot \mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}}} + \frac{4(q_G+1)^2+8}{2^\kappa - 1}.\end{aligned}$$

Proof. We assume without loss of generality that $\mathcal{A}_2$ does not query the challenge ciphertext c^* to its decryption oracle $\mathcal{O}_{SD}$ since doing so can only decrease the advantage. For $b_1, b_2 \in \{0,1\}$, we define the hybrid experiment H_{b_1,b_2} as follows: Let H_{b_1,b_2} be as $\mathsf{Exp}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}}$, but where the common reference string crs is obtained via $(crs, \tau) \leftarrow S_1(1^\kappa)$ (instead of an invocation of $\mathsf{NIZK.Gen}$). When $\mathcal{A}_1$ outputs (m_0, m_1, st), H_{b_1,b_2} computes c_1 as the encryption of m_{b_1} under ek_1, c_2 as the encryption of m_{b_2} under ek_2, and c_σ as in the real experiment (namely as the encryption of the two ElGamal public keys and the accompanying signature). It then simulates the proof π using S_2 and invokes $\mathcal{A}_2$ on input st and $c^* := (c_1, c_2, c_\sigma, \pi)$.

Claim 1. *There exist adversaries* $\mathcal{A}'_{\mathsf{ZK}}$ *and* $\mathcal{A}''_{\mathsf{ZK}}$ *such that*

$$\begin{aligned}\Pr^{H_{0,0}}[b' = 1] - \Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}}}[b' = 1 \mid b = 0] &= \mathsf{Adv}^{\mathsf{NIZK\text{-}ZK}}_{\mathsf{NIZK},\mathcal{A}'_{\mathsf{ZK}}}, \\ \Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}}}[b' = 1 \mid b = 1] - \Pr^{H_{1,1}}[b' = 1] &= \mathsf{Adv}^{\mathsf{NIZK\text{-}ZK}}_{\mathsf{NIZK},\mathcal{A}''_{\mathsf{ZK}}}.\end{aligned}$$

Proof of claim. We only prove the first part of the claim, the second one can be shown analogously. The adversary $\mathcal{A}'_{\mathsf{ZK}}$ on input crs, emulates toward $\mathcal{A}$ the experiment $H_{0,0}$. To this end, it generates all required keys. When generating the challenge ciphertext $c^* = (c_1, c_2, c_\sigma, \pi)$, it obtains π via the proof oracle. Note that in $H_{0,0}$, this ciphertext is a valid encryption of m_0, so the statement is correct and the proof oracle consequently returns a valid proof. When $\mathcal{A}$ returns a bit b', $\mathcal{A}'_{\mathsf{ZK}}$ returns $1 - b'$. Observe that if the CRS and the proofs are real, then this emulation is equivalent to the experiment $\mathsf{Exp}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}}$ when $b = 0$, and if the CRS and the proofs are simulated, then it is equivalent to $H_{0,0}$. Hence,

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{NIZK\text{-}ZK}}_{\mathsf{NIZK},\mathcal{A}'_{\mathsf{ZK}}} &= \Pr^{crs \leftarrow \mathsf{Gen}(1^\kappa)}\left[\mathcal{A}^{\mathsf{Prove}(crs,\cdot,\cdot)}(crs) = 1\right] - \Pr^{(crs,\tau)\leftarrow S_1(1^\kappa)}\left[\mathcal{A}^{S'(crs,\tau,\cdot,\cdot)}(crs) = 1\right] \\
&= 1 - \Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}}}[b' = 1 \mid b = 0] - \left(1 - \Pr^{H_{0,0}}[b' = 1]\right) \\
&= \Pr^{H_{0,0}}[b' = 1] - \Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}}}[b' = 1 \mid b = 0]. \qquad \Diamond
\end{aligned}$$

Analogous to the proof of Lemma 5.10, we next define three bad events. Let B_1 be the event that $\mathcal{A}$ queries its decryption oracle with a valid but improper ciphertext $(c_1, c_2, c_\sigma, \pi)$, i.e., $\left(g, ek^{\mathsf{PKE}}, vk^{\mathsf{Sig}}, c_1, c_2, c_\sigma\right) \notin L$, but where π is an accepting proof, i.e., $\mathsf{NIZK.Ver}\big(crs, x \coloneqq (g, ek^{\mathsf{PKE}}, vk^{\mathsf{Sig}}, c_1, c_2, c_\sigma), \pi\big) = 1$. As in the proof of Lemma 5.10, one can show that there exists an adversary $\mathcal{A}^{b_1,b_2}_{\mathsf{snd}}$ such that

$$\Pr^{H_{b_1,b_2}}[B_1] \leq q_{SD} \cdot \mathsf{Adv}^{\mathsf{NIZK\text{-}sim\text{-}snd}}_{\mathsf{NIZK},\mathcal{A}^{b_1,b_2}_{\mathsf{snd}}},$$

except that we here need (one-time) simulation soundness since the proof in the challenge ciphertext is simulated.

Further let B_2 be the event that $\mathcal{A}$ queries its decryption oracle with a valid and proper ciphertext $(c_1, c_2, c_\sigma, \pi)$, i.e., $\left(g, ek^{\mathsf{PKE}}, vk^{\mathsf{Sig}}, c_1, c_2, c_\sigma\right) \in L$ and π is accepting, but where c_σ is the encryption of a triple (ek_1, ek_2, σ), such that the pair (ek_1, ek_2) has never been output by the experiment or the oracle $\mathcal{O}_G$. Again as in the proof of Lemma 5.10, it can be shown that there exists an adversary $\mathcal{A}^{b_1,b_2}_{\mathsf{Sig}}$ such that

$$\Pr^{H_{b_1,b_2}}[B_2] \leq \mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}^{b_1,b_2}_{\mathsf{Sig}}}.$$

Finally, let B_3 be the event that H_{b_1,b_2} generates two different encryption keys $ek^{\mathsf{sPKE}} = \left(g, p, crs, ek^{\mathsf{PKE}}, vk^{\mathsf{Sig}}, ek_1, ek_2, \sigma\right)$ and $\left(ek^{\mathsf{sPKE}}\right)' = \left(g, p, crs, ek^{\mathsf{PKE}}, vk^{\mathsf{Sig}}, ek'_1, ek'_2, \sigma'\right)$ such that $ek_1 = ek'_1$ or $ek_2 = ek'_2$. Then,

$$\Pr^{H_{b_1,b_2}}[B_3] \leq \frac{2(q_G + 1)^2}{2^\kappa - 1},$$

which can be shown as in the proof of Lemma 5.10. For $B \coloneqq B_1 \cup B_2 \cup B_3$, we therefore have

$$\Pr^{H_{b_1,b_2}}[B] \leq q_{SD} \cdot \mathsf{Adv}^{\mathsf{NIZK\text{-}sim\text{-}snd}}_{\mathsf{NIZK},\mathcal{A}^{b_1,b_2}_{\mathsf{snd}}} + \mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}^{b_1,b_2}_{\mathsf{Sig}}} + \frac{2(q_G + 1)^2}{2^\kappa - 1}. \tag{9}$$

Claim 2. *There exist adversaries $\mathcal{A}'_{\mathsf{DDH}}$ and $\mathcal{A}''_{\mathsf{DDH}}$ such that*

$$\begin{aligned}
\Pr^{H_{1,0}}[b' = 1] - \Pr^{H_{0,0}}[b' = 1] &\leq 2 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{g,\mathcal{A}'_{\mathsf{DDH}}} + \Pr^{H_{0,0}}[B] + \Pr^{H_{1,0}}[B] + 2^{2-\kappa}, \\
\Pr^{H_{1,1}}[b' = 1] - \Pr^{H_{1,0}}[b' = 1] &\leq 2 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{g,\mathcal{A}''_{\mathsf{DDH}}} + 2^{2-\kappa}.
\end{aligned}$$

Proof of claim. We define the adversary $\mathcal{A}'_{\mathsf{DDH}}$ as follows. On input a triple (X, Y, T), it sets $ek_1 \leftarrow X$. It further generates all the remaining keys of the experiment (and thus lacks only the decryption key dk_1), samples $b \twoheadleftarrow \{0,1\}$, and emulates $H_{b,0}$ toward $\mathcal{A}$. The oracle $\mathcal{O}_{SD}$ is emulated by decrypting the second ciphertext component instead of the first one using dk_2. When $\mathcal{A}_1$ returns (m_0, m_1, st), $\mathcal{A}'_{\mathsf{DDH}}$ samples $r \twoheadleftarrow \mathbb{Z}_p^*$ and sets

$$c_1 \leftarrow (g^r, X^r, Y, T \cdot m_b).$$

It further computes c_2 as an ElGamal encryption of m_0, encrypts both keys and their signature to obtain c_σ, and simulates the NIZK proof π using S_2. It continues the emulation by giving $c^* := (c_1, c_2, c_\sigma, \pi)$ to $\mathcal{A}_2$. When $\mathcal{A}_2$ outputs its decision bit b', $\mathcal{A}'_{\mathsf{DDH}}$ outputs $d = 1$ if $b' = b$, and $d = 0$ otherwise.

First note that if (X, Y, T) are three uniform group elements, c_1 is independent of the bit b, and thus

$$\Pr^{\mathsf{DDH}^{\mathrm{rand}}_{g,\mathcal{A}'_{\mathsf{DDH}}}}[d = 1] = \frac{1}{2}.$$

On the other hand, if (X, Y, T) is a DDH triple, we have for a uniform $s \in \mathbb{Z}_p$,

$$c_1 = (g^r, X^r, Y, T \cdot m_b) = (g^r, ek_1^r, g^s, ek_1^s \cdot m_b),$$

which corresponds to a proper ElGamal encryption of m_b if $X \neq 1$ and $Y \neq 1$. Further note that, as in the proof of Lemma 5.10, $\mathcal{O}_{SD}$ is emulated perfectly if B does not occur. We therefore have

$$\Pr^{\mathsf{DDH}^{\mathrm{real}}_{g,\mathcal{A}'_{\mathsf{DDH}}}}[d = 1 \cap \neg B \mid X \neq 1 \neq Y] = \Pr^{H_{b,0}}[b' = b \cap \neg B].$$

This implies

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{DDH}}_{g,\mathcal{A}'_{\mathsf{DDH}}} &= \Pr^{\mathsf{DDH}^{\mathrm{real}}_{g,\mathcal{A}'_{\mathsf{DDH}}}}[d = 1] - \Pr^{\mathsf{DDH}^{\mathrm{rand}}_{g,\mathcal{A}'_{\mathsf{DDH}}}}[d = 1] \\
&\geq \Pr^{\mathsf{DDH}^{\mathrm{real}}_{g,\mathcal{A}'_{\mathsf{DDH}}}}[d = 1 \cap \neg B \mid X \neq 1 \neq Y] \cdot \Pr^{\mathsf{DDH}^{\mathrm{real}}_{g,\mathcal{A}'_{\mathsf{DDH}}}}[X \neq 1 \neq Y] - \frac{1}{2} \\
&= \Pr^{H_{b,0}}[b' = b \cap \neg B] \cdot \Pr^{\mathsf{DDH}^{\mathrm{real}}_{g,\mathcal{A}'_{\mathsf{DDH}}}}[X \neq 1 \neq Y] - \frac{1}{2}.
\end{aligned}$$

Using the union bound and $|G| = p \geq 2^\kappa$, we further have

$$\begin{aligned}
\Pr^{\mathsf{DDH}^{\mathrm{real}}_{g,\mathcal{A}'_{\mathsf{DDH}}}}[X \neq 1 \neq Y] &= 1 - \Pr^{\mathsf{DDH}^{\mathrm{real}}_{g,\mathcal{A}'_{\mathsf{DDH}}}}[X = 1 \vee Y = 1] \\
&\geq 1 - \Pr^{\mathsf{DDH}^{\mathrm{real}}_{g,\mathcal{A}'_{\mathsf{DDH}}}}[X = 1] - \Pr^{\mathsf{DDH}^{\mathrm{real}}_{g,\mathcal{A}'_{\mathsf{DDH}}}}[Y = 1] \\
&\geq 1 - 2 \cdot 2^{-\kappa}.
\end{aligned}$$

Hence,

$$\mathsf{Adv}^{\mathsf{DDH}}_{g,\mathcal{A}'_{\mathsf{DDH}}} \geq \Pr^{H_{b,0}}[b' = b \cap \neg B] - 2^{1-\kappa} - \frac{1}{2}.$$

Since

$$\Pr^{H_{b,0}}[b' = b] \leq \Pr^{H_{b,0}}[(b' = b \cap \neg B) \cup B] \leq \Pr^{H_{b,0}}[b' = b \cap \neg B] + \Pr^{H_{b,0}}[B],$$

we obtain

$$\begin{aligned}\mathsf{Adv}^{\mathsf{DDH}}_{g,\mathcal{A}'_{\mathsf{DDH}}} &\geq \Pr^{H_{b,0}}[b'=b] - \Pr^{H_{b,0}}[B] - 2^{1-\kappa} - \frac{1}{2} \\ &= \frac{1}{2}\Pr^{H_{0,0}}[b'=0] + \frac{1}{2}\Pr^{H_{1,0}}[b'=1] - \frac{1}{2}\Pr^{H_{0,0}}[B] - \frac{1}{2}\Pr^{H_{1,0}}[B] - 2^{1-\kappa} - \frac{1}{2} \\ &= \frac{1}{2}\Pr^{H_{1,0}}[b'=1] - \frac{1}{2}\Pr^{H_{0,0}}[b'=1] - \frac{1}{2}\Pr^{H_{0,0}}[B] - \frac{1}{2}\Pr^{H_{1,0}}[B] - 2^{1-\kappa}.\end{aligned}$$

Rearranging the inequality concludes the proof of the first part of the claim.

The second part of the claim can be proven analogously, where $\mathcal{A}''_{\mathsf{DDH}}$ sets $ek_2 \leftarrow X$ instead of $ek_1 \leftarrow X$. Since it therefore has dk_1, which is the key used by the decryption algorithm, the decryption oracle can be emulated perfectly, even if B occurs. ◊

Using Claims 1 and 2, we get

$$\begin{aligned}\mathsf{Adv}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}} &\leq 2 \cdot \Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}}}[b'=b] - 1 \\ &= 2\left(\frac{1}{2} \cdot \Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}}}[b'=0 \mid b=0] + \frac{1}{2} \cdot \Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}}}[b'=1 \mid b=1]\right) - 1 \\ &= \Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}}}[b'=1 \mid b=1] - \Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}}}[b'=1 \mid b=0] \\ &= \Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}}}[b'=1 \mid b=1] - \Pr^{H_{1,1}}[b'=1] \\ &\quad + \Pr^{H_{1,1}}[b'=1] - \Pr^{H_{1,0}}[b'=1] + \Pr^{H_{1,0}}[b'=1] - \Pr^{H_{0,0}}[b'=1] \\ &\quad + \Pr^{H_{0,0}}[b'=1] - \Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}}}[b'=1 \mid b=0] \\ &\leq \mathsf{Adv}^{\mathsf{NIZK\text{-}ZK}}_{\mathsf{NIZK},\mathcal{A}'_{\mathsf{ZK}}} + \mathsf{Adv}^{\mathsf{NIZK\text{-}ZK}}_{\mathsf{NIZK},\mathcal{A}''_{\mathsf{ZK}}} + 2 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{g,\mathcal{A}'_{\mathsf{DDH}}} + 2 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{g,\mathcal{A}''_{\mathsf{DDH}}} \\ &\quad + \Pr^{H_{0,0}}[B] + \Pr^{H_{1,0}}[B] + 2^{3-\kappa}.\end{aligned}$$

Let $\mathcal{A}_{\mathsf{ZK}}$ be the adversary that runs $\mathcal{A}'_{\mathsf{ZK}}$ and $\mathcal{A}''_{\mathsf{ZK}}$ with probability 1/2 each, let $\mathcal{A}_{\mathsf{DDH}}$ run $\mathcal{A}'_{\mathsf{DDH}}$ and $\mathcal{A}''_{\mathsf{DDH}}$ with probability 1/2 each, let $\mathcal{A}_{\mathsf{snd}}$ run $\mathcal{A}^{0,0}_{\mathsf{snd}}$ and $\mathcal{A}^{1,0}_{\mathsf{snd}}$ with probability 1/2 each, and let $\mathcal{A}_{\mathsf{Sig}}$ run $\mathcal{A}^{0,0}_{\mathsf{Sig}}$ and $\mathcal{A}^{1,0}_{\mathsf{Sig}}$ with probability 1/2 each. Combing the result above with equation (9), we can then conclude

$$\begin{aligned}\mathsf{Adv}^{\mathsf{sPKE\text{-}IND\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}} &\leq 4 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{g,\mathcal{A}_{\mathsf{DDH}}} + 2 \cdot \mathsf{Adv}^{\mathsf{NIZK\text{-}ZK}}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}}} + 2q_{SD} \cdot \mathsf{Adv}^{\mathsf{NIZK\text{-}sim\text{-}snd}}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{snd}}} \\ &\quad + 2 \cdot \mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}}} + \frac{4(q_G+1)^2+8}{2^\kappa - 1}.\end{aligned}$$

□

We finally show that sPKE is IK-CCA secure.

Lemma C.2. *Let* sPKE *be the scheme from Section 5.2 and let* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *be a pair of probabilistic algorithms such that* $\mathcal{A}_1$ *and* $\mathcal{A}_2$ *together make at most* q_G *queries to* $\mathcal{O}_G$ *and at most* q_{SD} *queries to* $\mathcal{O}_{SD_0}$ *and* $\mathcal{O}_{SD_1}$ *combined. Then, there exist adversaries* $\mathcal{A}_{\mathsf{DDH}}$, $\mathcal{A}_{\mathsf{ZK}}$, $\mathcal{A}_{\mathsf{snd}}$, $\mathcal{A}_{\mathsf{PKE}}$, *and* $\mathcal{A}_{\mathsf{Sig}}$ *(which are all roughly as efficient as emulating an execution of* $\mathsf{Exp}^{\mathsf{sPKE\text{-}IK\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}}$*) such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{sPKE\text{-}IK\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}} &\leq 8 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{g,\mathcal{A}_{\mathsf{DDH}}} + 2 \cdot \mathsf{Adv}^{\mathsf{NIZK\text{-}ZK}}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}}} + 8q_{SD} \cdot \mathsf{Adv}^{\mathsf{NIZK\text{-}sim\text{-}snd}}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{snd}}} \\ &\quad + 2 \cdot \mathsf{Adv}^{\mathsf{PKE\text{-}IND\text{-}CPA}}_{\mathsf{PKE},\mathcal{A}_{\mathsf{PKE}}} + 8 \cdot \mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}}} + \frac{16(q_G+2)^2+32}{2^\kappa - 1}.\end{aligned}$$

Proof. We assume without loss of generality that $\mathcal{A}_2$ does not query the challenge ciphertext c^* to any of its decryption oracles $\mathcal{O}_{SD_0}$ or $\mathcal{O}_{SD_1}$, since doing so can only decrease the advantage. We define hybrid experiments H_0 to H_5 as follows:

- H_0 is identical to $\mathsf{Exp}^{\mathsf{sPKE\text{-}IK\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}}$, except that the common reference string crs is obtained via $(crs, \tau) \leftarrow S_1(1^\kappa)$ (instead of an invocation of $\mathsf{NIZK.Gen}$), and the proof π in the challenge ciphertext c^* is simulated using S_2.
- H_1 is identical to H_0, but when $\mathcal{A}_1$ outputs (m, st), the hybrid computes c_σ not as an encryption of $(ek_{b,1}, ek_{b,2}, \sigma_b)$, but as the encryption of 0^ℓ, where ℓ is the length of the encoding of $(ek_{b,1}, ek_{b,2}, \sigma_b)$ (where the encoding needs to be chosen such that this length is equal for all keys).
- H_2 is identical to H_1, except that for the generation of the challenge ciphertext c^*, the key $ek_{0,1}$ is replaced by $g^{d_{0,1}}$ for a freshly sampled $d_{0,1} \twoheadleftarrow \mathbb{Z}_p^*$.
- H_3 is identical to H_2, except that for the generation of the challenge ciphertext c^*, the key $ek_{0,2}$ is replaced by $g^{d_{0,2}}$ for a freshly sampled $d_{0,2} \twoheadleftarrow \mathbb{Z}_p^*$.
- H_4 is identical to H_3, except that for the generation of the challenge ciphertext c^*, the key $ek_{1,1}$ is replaced by $g^{d_{1,1}}$ for a freshly sampled $d_{1,1} \twoheadleftarrow \mathbb{Z}_p^*$.
- H_5 is identical to H_4, except that for the generation of the challenge ciphertext c^*, the key $ek_{1,2}$ is replaced by $g^{d_{1,2}}$ for a freshly sampled $d_{1,2} \twoheadleftarrow \mathbb{Z}_p^*$.

Note that the view of $\mathcal{A}$ in H_5 is independent from the bit b, which implies

$$\Pr^{H_5}[b' = b] = \frac{1}{2}. \tag{10}$$

It can be shown as in the proof of Lemma C.1 that there exist an adversary $\mathcal{A}_{\mathsf{ZK}}$ such that

$$\Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}IK\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}}}[b' = b] - \Pr^{H_0}[b' = b] = \mathsf{Adv}^{\mathsf{NIZK\text{-}ZK}}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}}}. \tag{11}$$

Claim 1. *There exists an adversary $\mathcal{A}_{\mathsf{PKE}}$ such that*

$$\Pr^{H_0}[b' = b] - \Pr^{H_1}[b' = b] = \mathsf{Adv}^{\mathsf{PKE\text{-}IND\text{-}CPA}}_{\mathsf{PKE},\mathcal{A}_{\mathsf{PKE}}}.$$

Proof of claim. When $\mathcal{A}_{\mathsf{PKE}}$ obtains a public key ek from the CPA challenger, it generates all remaining keys itself and emulates H_0 (or H_1) toward $\mathcal{A}$. Note that $\mathcal{A}_{\mathsf{PKE}}$ never needs to decrypt any of the ciphertexts c_σ in the experiment and thus, the missing decryption key is not needed for the emulation. When $\mathcal{A}_1$ outputs (m, st), $\mathcal{A}_{\mathsf{PKE}}$ gives $\big(0^\ell, (ek_{b,1}, ek_{b,2}, \sigma_b)\big)$ to its CPA challenger to obtain a ciphertext c_σ, where ℓ is the length of the encoding of $(ek_{b,1}, ek_{b,2}, \sigma_b)$. The rest is done as in H_0. When $\mathcal{A}_2$ returns a bit b', $\mathcal{A}_{\mathsf{PKE}}$ returns $b'' = 1$ if $b' = b$, and $b'' = 0$ if $b' \neq b$.

Note that if the CPA challenger chooses the bit $b_{\mathsf{CPA}} = 0$, c_σ is an encryption of 0^ℓ, as in H_1, and if $b_{\mathsf{CPA}} = 1$, c_σ is as in H_0. Hence,

$$\Pr^{\mathsf{Exp}^{\mathsf{PKE\text{-}IND\text{-}CPA}}_{\mathsf{PKE},\mathcal{A}_{\mathsf{PKE}}}}[b'' = 1 \mid b_{\mathsf{CPA}} = 0] = \Pr^{H_1}[b' = b],$$
$$\Pr^{\mathsf{Exp}^{\mathsf{PKE\text{-}IND\text{-}CPA}}_{\mathsf{PKE},\mathcal{A}_{\mathsf{PKE}}}}[b'' = 1 \mid b_{\mathsf{CPA}} = 1] = \Pr^{H_0}[b' = b].$$

We can therefore conclude

$$\begin{aligned}\mathsf{Adv}^{\mathsf{PKE\text{-}IND\text{-}CPA}}_{\mathsf{PKE},\mathcal{A}_{\mathsf{PKE}}} &= 2 \cdot \Pr^{\mathsf{Exp}^{\mathsf{PKE\text{-}IND\text{-}CPA}}_{\mathsf{PKE},\mathcal{A}_{\mathsf{PKE}}}}[b'' = b_{\mathsf{CPA}}] - 1 \\ &= \Pr^{\mathsf{Exp}^{\mathsf{PKE\text{-}IND\text{-}CPA}}_{\mathsf{PKE},\mathcal{A}_{\mathsf{PKE}}}}[b'' = 0 \mid b_{\mathsf{CPA}} = 0] + \Pr^{\mathsf{Exp}^{\mathsf{PKE\text{-}IND\text{-}CPA}}_{\mathsf{PKE},\mathcal{A}_{\mathsf{PKE}}}}[b'' = 1 \mid b_{\mathsf{CPA}} = 1] - 1 \\ &= \Pr^{\mathsf{Exp}^{\mathsf{PKE\text{-}IND\text{-}CPA}}_{\mathsf{PKE},\mathcal{A}_{\mathsf{PKE}}}}[b'' = 1 \mid b_{\mathsf{CPA}} = 1] - \Pr^{\mathsf{Exp}^{\mathsf{PKE\text{-}IND\text{-}CPA}}_{\mathsf{PKE},\mathcal{A}_{\mathsf{PKE}}}}[b'' = 1 \mid b_{\mathsf{CPA}} = 0] \\ &= \Pr^{H_0}[b' = b] - \Pr^{H_1}[b' = b]. \qquad \Diamond\end{aligned}$$

We define the event B analogous to the events in the proof of Lemmata 5.10 and C.1. As there, one can show for $i \in \{0, \ldots, 5\}$ that there exist adversaries $\mathcal{A}^i_{\mathsf{snd}}$ and $\mathcal{A}^i_{\mathsf{Sig}}$ such that

$$\Pr^{H_i}[B] \leq q_{SD} \cdot \mathsf{Adv}^{\mathsf{NIZK\text{-}sim\text{-}snd}}_{\mathsf{NIZK},\mathcal{A}^i_{\mathsf{snd}}} + \mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}^i_{\mathsf{Sig}}} + \frac{2(q_G+2)^2}{2^\kappa - 1}. \tag{12}$$

Claim 2. *There exist adversaries $\mathcal{A}^1_{\mathsf{DDH}}, \ldots, \mathcal{A}^4_{\mathsf{DDH}}$ such that for $i \in \{1, 3\}$*

$$\Pr^{H_i}[b' = b] - \Pr^{H_{i+1}}[b' = b] \leq \mathsf{Adv}^{\mathsf{DDH}}_{g,\mathcal{A}^i_{\mathsf{DDH}}} + \Pr^{H_i}[B] + \Pr^{H_{i+1}}[B] + 2^{2-\kappa},$$

and for $i \in \{2, 4\}$

$$\Pr^{H_i}[b' = b] - \Pr^{H_{i+1}}[b' = b] \leq \mathsf{Adv}^{\mathsf{DDH}}_{g,\mathcal{A}^i_{\mathsf{DDH}}} + 2^{2-\kappa}.$$

Proof of claim. On input (X, Y, T), $\mathcal{A}^1_{\mathsf{DDH}}$ sets $ek_{0,1} \leftarrow X$, $\mathcal{A}^2_{\mathsf{DDH}}$ sets $ek_{0,2} \leftarrow X$, $\mathcal{A}^3_{\mathsf{DDH}}$ sets $ek_{1,1} \leftarrow X$, and $\mathcal{A}^4_{\mathsf{DDH}}$ sets $ek_{1,2} \leftarrow X$. All adversaries generate the remaining keys themselves and emulate H_i (or H_{i+1}) toward $\mathcal{A}$. To emulate the decryption oracles, $\mathcal{A}^1_{\mathsf{DDH}}$ and $\mathcal{A}^3_{\mathsf{DDH}}$ decrypt the second ciphertext component instead of the first one; $\mathcal{A}^2_{\mathsf{DDH}}$ and $\mathcal{A}^4_{\mathsf{DDH}}$ can emulate all oracles perfectly. As in the proof of Lemma 5.10, $\mathcal{O}_{SD}$ is also emulated perfectly by $\mathcal{A}^1_{\mathsf{DDH}}$ and $\mathcal{A}^3_{\mathsf{DDH}}$ if the event B does not occur. When $\mathcal{A}_1$ returns (m, st) and if $b = 0$, then $\mathcal{A}^1_{\mathsf{DDH}}$ samples $r \twoheadleftarrow \mathbb{Z}^*_p$, sets

$$c_1 \leftarrow \big(Y^r, T^r, Y, T \cdot m\big),$$

and generates the remaining ciphertext components as in the real experiment. The other adversaries generate the ciphertext components analogously. When $\mathcal{A}_2$ returns a bit b', then $\mathcal{A}^i_{\mathsf{DDH}}$ returns $d = 1$ if $b' = b$ and $d = 0$ if $b' \neq b$ for all $i \in \{1, \ldots, 4\}$.

Consider the case $i = 1$ and note that if (X, Y, T) is a DDH triple, we have $Y = g^s$ and $T = X^s$ for a uniform $s \in \mathbb{Z}_p$, and thus, if $b = 0$,

$$c_1 = \big(Y^r, X^{s \cdot r}, Y, X^s \cdot m\big) = \big(g^{s \cdot r}, (ek_{0,1})^{s \cdot r}, g^s, (ek_{0,1})^s \cdot m\big).$$

If $X \neq 1 \neq Y$, this corresponds to an encryption of m under $ek_{0,1}$, as in H_1. On the other hand, if (X, Y, T) are three uniform group elements and $X \neq 1 \neq Y$, then there are (uniformly distributed) $s, d_{0,1} \in \mathbb{Z}^*_p$ such that $Y = g^s$ and $T = g^{s \cdot d_{0,1}}$. Hence, we have in this case

$$c_1 = \big(g^{s \cdot r}, g^{s \cdot d_{0,1} \cdot r}, g^s, g^{s \cdot d_{0,1}} \cdot m\big) = \big(g^{s \cdot r}, \big(g^{d_{0,1}}\big)^{s \cdot r}, g^s, \big(g^{d_{0,1}}\big)^s \cdot m\big),$$

which corresponds to an encryption under the fresh key $g^{d_{0,1}}$, as in H_2. Further note that if $b = 1$, the emulation, H_1, and H_2 are all equivalent. We therefore have

$$\begin{aligned}\Pr^{\mathsf{DDH}^{\mathrm{real}}_{g,\mathcal{A}^1_{\mathsf{DDH}}}}[d = 1 \cap \neg B \mid X \neq 1 \neq Y] &= \Pr^{H_1}[b' = b \cap \neg B], \\ \Pr^{\mathsf{DDH}^{\mathrm{rand}}_{g,\mathcal{A}^1_{\mathsf{DDH}}}}[d = 1 \cap \neg B \mid X \neq 1 \neq Y] &= \Pr^{H_2}[b' = b \cap \neg B].\end{aligned}$$

As in the proof of Lemma C.1, we obtain

$$\Pr^{\mathsf{DDH}^{\mathrm{real}}_{g,\mathcal{A}^1_{\mathsf{DDH}}}}[d=1] \geq \Pr^{H_1}[b'=b] - \Pr^{H_1}[B] - 2^{1-\kappa}.$$

Moreover,

$$\begin{aligned}
\Pr^{\mathsf{DDH}^{\mathrm{rand}}_{g,\mathcal{A}^1_{\mathsf{DDH}}}}[d=1] &= \Pr^{\mathsf{DDH}^{\mathrm{rand}}_{g,\mathcal{A}^1_{\mathsf{DDH}}}}[d=1 \cap \neg B \mid X \neq 1 \neq Y] \cdot \Pr^{\mathsf{DDH}^{\mathrm{rand}}_{g,\mathcal{A}^1_{\mathsf{DDH}}}}[X \neq 1 \neq Y] \\
&\quad + \Pr^{\mathsf{DDH}^{\mathrm{rand}}_{g,\mathcal{A}^1_{\mathsf{DDH}}}}[d=1 \cap (B \cup [X=1 \vee Y=1])] \\
&\leq \Pr^{H_2}[b'=b \cap \neg B] + \Pr^{\mathsf{DDH}^{\mathrm{rand}}_{g,\mathcal{A}^1_{\mathsf{DDH}}}}[B] + \Pr^{\mathsf{DDH}^{\mathrm{rand}}_{g,\mathcal{A}^1_{\mathsf{DDH}}}}[X=1 \vee Y=1] \\
&\leq \Pr^{H_2}[b'=b] + \Pr^{\mathsf{DDH}^{\mathrm{rand}}_{g,\mathcal{A}^1_{\mathsf{DDH}}}}[B] + 2^{1-\kappa}.
\end{aligned}$$

Since the probability of B in $\mathsf{DDH}^{\mathrm{rand}}_{g,\mathcal{A}^1_{\mathsf{DDH}}}$ is equal to its probability in H_2, we conclude

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{DDH}}_{g,\mathcal{A}^1_{\mathsf{DDH}}} &= \Pr^{\mathsf{DDH}^{\mathrm{real}}_{g,\mathcal{A}^1_{\mathsf{DDH}}}}[d=1] - \Pr^{\mathsf{DDH}^{\mathrm{rand}}_{g,\mathcal{A}^1_{\mathsf{DDH}}}}[d=1] \\
&\geq \Pr^{H_1}[b'=b] - \Pr^{H_2}[b'=b] - \Pr^{H_1}[B] - \Pr^{H_2}[B] - 2^{2-\kappa}.
\end{aligned}$$

The proofs for $i \in \{2,3,4\}$ are analogous, where for $i \in \{2,4\}$, the occurrence of B does not matter since the decryption oracle can always be emulated perfectly. ◊

Using equation (11), Claims 1 and 2, and equation (10), we obtain

$$\begin{aligned}
\Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}IK\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}}}[b'=b] &= \Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}IK\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}}}[b'=b] - \Pr^{H_0}[b'=b] + \Pr^{H_0}[b'=b] - \Pr^{H_1}[b'=b] \\
&\quad + \sum_{i=1}^{4}\Big(\Pr^{H_i}[b'=b] - \Pr^{H_{i+1}}[b'=b]\Big) + \Pr^{H_5}[b'=b] \\
&\leq \mathsf{Adv}^{\mathsf{NIZK\text{-}ZK}}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}}} + \mathsf{Adv}^{\mathsf{PKE\text{-}IND\text{-}CPA}}_{\mathsf{PKE},\mathcal{A}_{\mathsf{PKE}}} + \sum_{i=1}^{4}\Big(\mathsf{Adv}^{\mathsf{DDH}}_{g,\mathcal{A}^i_{\mathsf{DDH}}} + \Pr^{H_i}[B]\Big) + 2^{4-\kappa} + \frac{1}{2}.
\end{aligned}$$

For the adversary $\mathcal{A}_{\mathsf{snd}}$ that runs $\mathcal{A}^1_{\mathsf{snd}}, \ldots, \mathcal{A}^4_{\mathsf{snd}}$ with probability 1/4 each, and the adversary $\mathcal{A}_{\mathsf{Sig}}$ that runs $\mathcal{A}^1_{\mathsf{Sig}}, \ldots, \mathcal{A}^4_{\mathsf{Sig}}$ with probability 1/4 each, we obtain using equation (12),

$$\sum_{i=1}^{4} \Pr^{H_i}[B] \leq 4q_{SD} \cdot \mathsf{Adv}^{\mathsf{NIZK\text{-}sim\text{-}snd}}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{snd}}} + 4 \cdot \mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}}} + \frac{8(q_G+2)^2}{2^\kappa - 1}.$$

Further defining $\mathcal{A}_{\mathsf{DDH}}$ as running $\mathcal{A}^1_{\mathsf{DDH}}, \ldots, \mathcal{A}^4_{\mathsf{DDH}}$ with probability 1/4 each yields

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{sPKE\text{-}IK\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}} &\leq 2 \cdot \Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}IK\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}}}[b'=b] - 1 \\
&\leq 2 \cdot \mathsf{Adv}^{\mathsf{NIZK\text{-}ZK}}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}}} + 2 \cdot \mathsf{Adv}^{\mathsf{PKE\text{-}IND\text{-}CPA}}_{\mathsf{PKE},\mathcal{A}_{\mathsf{PKE}}} + 8 \cdot \mathsf{Adv}^{\mathsf{DDH}}_{g,\mathcal{A}_{\mathsf{DDH}}} + 8q_{SD} \cdot \mathsf{Adv}^{\mathsf{NIZK\text{-}sim\text{-}snd}}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{snd}}} \\
&\quad + 8 \cdot \mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}}} + \frac{16(q_G+2)^2}{2^\kappa - 1} + 2^{5-\kappa}.
\end{aligned}$$

Observing that $2^{5-\kappa} \leq \frac{32}{2^\kappa - 1}$ concludes the proof. □

D Remaining Security Proofs of the ACE Scheme for Equality

After privacy, which was shown in Lemma 6.3, we prove anonymity, which can be shown similarly. We provide a proof for strong anonymity. Note, however, that for the equality policy, strong anonymity does not provide more guarantees than weak anonymity because anyone who can decrypt directly learns that the sender role is equal to the receiver role.

Lemma D.1. *Let* ACE *be the scheme from above, let* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *be an attacker on the anonymity such that* $\mathcal{A}_1$ *makes at most* q_S *queries of the form* $(\cdot, \mathtt{sen})$ *to the oracle* $\mathcal{O}_G$*, and at most* q_D *queries to* $\mathcal{O}_{SD}$*. Then, there exist probabilistic algorithms* $\mathcal{A}_{\mathsf{PRF}}$*,* $\mathcal{A}_{\mathsf{ZK}}$*, and* $\mathcal{A}_{\mathsf{sPKE}}$ *(which are all roughly as efficient as emulating an execution of* $\mathsf{Exp}^{\text{ACE-PRV-ANON-CCA}}_{\mathsf{ACE},\mathcal{A}}$*) such that*

$$\mathsf{Adv}^{\text{ACE-sANON-CCA}}_{\mathsf{ACE},\mathcal{A}} \leq 2 \cdot \mathsf{Adv}^{\mathsf{PRF}}_{F,\mathcal{A}_{\mathsf{PRF}}} + 2 \cdot \mathsf{Adv}^{\text{NIZK-ZK}}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}}} + (q_S + q_D + 1)^2 \cdot \mathsf{Adv}^{\text{sPKE-IK-CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}.$$

Proof. We assume without loss of generality that $\mathcal{A}$ ensures $m_0 = m_1$ and $P(i_0, j) = P(i_1, j)$ for all $j \in J$, since doing otherwise can only decrease the advantage. Since we have $P(i, j) = 1 \Leftrightarrow i = j$, the latter condition implies that if $i_0 \in J$ or $i_1 \in J$, then $i_0 = i_1$. In case $i_0 = i_1$ and $m_0 = m_1$, $\mathcal{A}$ cannot have positive advantage. Hence, we can further assume without loss of generality that $i_0 \notin J$ and $i_1 \notin J$. As in the proof of Lemma 6.3, let $H_0 \coloneqq \mathsf{Exp}^{\text{ACE-PRV-ANON-CCA}}_{\mathsf{ACE},\mathcal{A}}$, let H_1 be as H_0 where F_K is replaced by a truly uniform random function U, and let H_2 be as H_1, where $crs^{\mathsf{NIZK}} \leftarrow \mathsf{NIZK.Gen}(1^\kappa)$ in ACE.Setup is replaced by $\big(crs^{\mathsf{NIZK}}, \tau^{\mathsf{NIZK}}\big) \leftarrow S_1^{\mathsf{NIZK}}(1^\kappa)$ and for the generation of the challenge ciphertext c^*, $\pi^{\mathsf{NIZK}} \leftarrow \mathsf{NIZK.Prove}\big(crs^{\mathsf{NIZK}}, x, w\big)$ in ACE.Enc is replaced by $\pi^{\mathsf{NIZK}} \leftarrow S_2^{\mathsf{NIZK}}\big(crs^{\mathsf{NIZK}}, \tau^{\mathsf{NIZK}}, x\big)$. An identical proof as the one in the proof of Lemma 6.3 shows that there exist $\mathcal{A}_{\mathsf{PRF}}$ and $\mathcal{A}_{\mathsf{ZK}}$ such that

$$\Pr^{H_0}\big[b' = b\big] - \Pr^{H_2}\big[b' = b\big] = \mathsf{Adv}^{\mathsf{PRF}}_{F,\mathcal{A}_{\mathsf{PRF}}} + \mathsf{Adv}^{\text{NIZK-ZK}}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}}}.$$

We now transform $\mathcal{A}$ to a winner $\mathcal{A}_{\mathsf{sPKE}}$ for the anonymity game for the scheme sPKE. The reduction is similar to the one in the proof of Lemma 6.3, but $\mathcal{A}_{\mathsf{sPKE}}$ has to guess both i_0 and i_1, which is why we loose the quadratic factor $(q_S + q_D + 1)^2$. On input $(sp^{\mathsf{sPKE}}, ek_0^{\mathsf{sPKE}}, ek_1^{\mathsf{sPKE}})$, $\mathcal{A}_{\mathsf{sPKE}}$ initializes $i_{q_0}, i_{q_1} \leftarrow \perp$, $k_q \leftarrow 1$, chooses $q_0, q_1 \twoheadleftarrow \{0, \ldots, q_S + q_D\}$ uniformly at random, runs $\big(vk^{\mathsf{Sig}}, sk^{\mathsf{Sig}}\big) \leftarrow \mathsf{Sig.Gen}(1^\kappa)$, and $\big(crs^{\mathsf{NIZK}}, \tau^{\mathsf{NIZK}}\big) \leftarrow S_1^{\mathsf{NIZK}}(1^\kappa)$, and gives $sp^{\mathsf{ACE}} \coloneqq \big(sp^{\mathsf{sPKE}}, vk^{\mathsf{Sig}}, crs^{\mathsf{NIZK}}\big)$ to $\mathcal{A}_1$. It emulates the oracles for $\mathcal{A}_1$ as follows.

$\mathcal{O}_G(\cdot,\cdot)$**:** On query $(i, \mathtt{sen})$, if $k_q \notin \{q_0, q_1\}$ and $i \notin \{i_{q_0}, i_{q_1}\}$, then generate an encryption key $ek_i^{\mathsf{ACE}} \coloneqq \big(vk^{\mathsf{Sig}}, ek_i^{\mathsf{sPKE}}, vk_i^{\mathsf{Sig}}, sk_i^{\mathsf{Sig}}, \sigma_i^{\mathsf{Sig}}, crs^{\mathsf{NIZK}}\big)$ as H_2 does, where $\big(ek_i^{\mathsf{sPKE}}, dk_i^{\mathsf{sPKE}}\big)$ is obtained via $\mathcal{O}_G$ and remembered for future queries. If $k_q = q_l$ or $i = i_{q_l}$ for some $l \in \{0, 1\}$, replace ek_i^{sPKE} by ek_l^{sPKE} (by ek_0^{sPKE} if $q_0 = q_1$) and set $i_{q_l} \leftarrow i$. In both cases, set $k_q \leftarrow k_q + 1$ at the end. On query $(j, \mathtt{rec})$, obtain a decryption key from $\mathcal{O}_G$ and remember it for later.

$\mathcal{O}_{SD}(\cdot,\cdot)$**:** On query $\big(j, c = \big(\tilde{c}, \pi^{\mathsf{NIZK}}\big)\big)$, if $k_q \notin \{q_0, q_1\}$ and $j \notin \{i_{q_0}, i_{q_1}\}$, then execute $c' \leftarrow \mathsf{ACE.San}(sp^{\mathsf{ACE}}, c)$, generate a decryption key dk_j^{ACE} as above, decrypt c' using dk_j^{ACE}, and return the resulting message. If $k_q = q_l$ or $j = i_{q_l}$ for some $l \in \{0, 1\}$, set $i_{q_l} \leftarrow j$ and use the oracle $\mathcal{O}_{SD_l}$ of the IK-CCA experiment to obtain a decryption m of $\tilde{c}$. If $\mathsf{NIZK.Ver}\big(crs^{\mathsf{NIZK}}, x \coloneqq \big(vk^{\mathsf{Sig}}, \tilde{c}, \big), \pi^{\mathsf{NIZK}}\big) = 1$, return m, otherwise, return $\perp$. In all cases, set $k_q \leftarrow k_q + 1$ at the end.

When $\mathcal{A}_1$ returns (m_0, m_1, i_0, i_1, st), $\mathcal{A}_{\mathsf{sPKE}}$ outputs m_0 to the challenger of the anonymity experiment to obtain a challenge ciphertext $\tilde{c}^*$. It then runs $S_2^{\mathsf{NIZK}}\big(crs^{\mathsf{NIZK}}, \tau^{\mathsf{NIZK}}, x := \big(vk^{\mathsf{Sig}}, \tilde{c}^*\big)\big)$, and gives st and the ciphertext $c^* := \big(\tilde{c}^*, \pi^{\mathsf{NIZK}}\big)$ to $\mathcal{A}_2$. It emulates the oracles for $\mathcal{A}_2$ as follows:

$\mathcal{O}_G(\cdot,\cdot)$**:** On query $(i, \mathtt{sen})$, if $i \notin \{i_0, i_1\}$, then generate an encryption key $ek_i^{\mathsf{ACE}} := \big(vk^{\mathsf{Sig}}, ek_i^{\mathsf{sPKE}}, vk_i^{\mathsf{Sig}}, sk_i^{\mathsf{Sig}}, \sigma_i^{\mathsf{Sig}}, crs^{\mathsf{NIZK}}\big)$ as H_2 does, where $\big(ek_i^{\mathsf{sPKE}}, dk_i^{\mathsf{sPKE}}\big)$ is obtained via $\mathcal{O}_G$ and remembered for future queries. If $i = i_{q_l}$ for some $l \in \{0,1\}$, replace ek_i^{sPKE} by ek_l^{sPKE}. On query $(j, \mathtt{rec})$, obtain a decryption key as before.

$\mathcal{O}_{SD^*}(\cdot,\cdot)$**:** On query $\big(j, c = \big(\tilde{c}, \pi^{\mathsf{NIZK}}\big)\big)$, run $\mathsf{ACE.DMod}(sp^{\mathsf{ACE}}, c^*, c)$. If the output is 1, return $\mathtt{test}$. Otherwise, if $j \notin \{i_0, i_1\}$, run $c' \leftarrow \mathsf{ACE.San}(sp^{\mathsf{ACE}}, c)$, generate a decryption key dk_j^{ACE} as above, decrypt c' using dk_j^{ACE}, and return the resulting message. If $j = i_{q_l}$ for some $l \in \{0,1\}$, use the oracle $\mathcal{O}_{SD_l}$ of the IK-CCA experiment to obtain a decryption m of $\tilde{c}$. If $\mathsf{NIZK.Ver}\big(crs^{\mathsf{NIZK}}, x := \big(vk^{\mathsf{Sig}}, \tilde{c}, \big), \pi^{\mathsf{NIZK}}\big) = 1$, return m, otherwise, return $\perp$.

Note that $\mathcal{A}_{\mathsf{sPKE}}$ never queries any of the decryption oracles of the IK-CCA experiment on $\tilde{c}^*$ because we return $\mathtt{test}$ whenever this would be necessary. Denote by Q the event that for all $l \in \{0,1\}$ we have either $i_{q_l} = i_l$, or $q_l = 0$ and $\mathcal{A}_1$ does not make the query $(i_l, \mathtt{sen})$ to $\mathcal{O}_G$ and no queries for role i_l to $\mathcal{O}_{SD}$. When $\mathcal{A}_2$ returns a bit b' and Q holds, $\mathcal{A}_{\mathsf{sPKE}}$ returns the same bit $b'' \leftarrow b'$, if $\neg Q$, $\mathcal{A}_{\mathsf{sPKE}}$ returns a uniform bit $b'' \twoheadleftarrow \{0,1\}$.

Let $\tilde{b}$ be the bit chosen by the IK-CCA experiment. Note that if Q occurs, the view of $\mathcal{A}$ is identical to the one in H_2 with $b = \tilde{b}$. This implies

$$\Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}IK\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}\big[b'' = \tilde{b} \;\big|\; Q\big] = \Pr^{H_2}\big[b' = b\big].$$

Using that the probability of Q is $1/(q_S + q_D + 1)^2$, it follows as in the proof of Lemma 6.3 that

$$\mathsf{Adv}^{\mathsf{ACE\text{-}sANON\text{-}CCA}}_{\mathsf{ACE},\mathcal{A}} = 2 \cdot \mathsf{Adv}^{\mathsf{PRF}}_{F,\mathcal{A}_{\mathsf{PRF}}} + 2 \cdot \mathsf{Adv}^{\mathsf{NIZK\text{-}ZK}}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}}} + (q_S + q_D + 1)^2 \cdot \mathsf{Adv}^{\mathsf{sPKE\text{-}IK\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}. \qquad \square$$

We next prove sanitization security of our scheme.

Lemma D.2. *Let* ACE *be the scheme from above, and let* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *be an attacker on the sanitization security such that* $\mathcal{A}_1$ *makes at most* q_{S_1} *queries of the form* $(\cdot, \mathtt{sen})$ *and at most* q_{R_1} *queries of the form* $(\cdot, \mathtt{rec})$ *to the oracle* $\mathcal{O}_G$*, and at most* q_{D_1} *queries to* $\mathcal{O}_{SD}$*, and* $\mathcal{A}_2$ *makes at most* q_{R_2} *queries of the form* $(\cdot, \mathtt{rec})$ *to the oracle* $\mathcal{O}_G$*. Then, there exist probabilistic algorithms* $\mathcal{A}_{\mathsf{PRF}}$*,* $\mathcal{A}_{\mathsf{ZK}_1}$*,* $\mathcal{A}_{\mathsf{ZK}_2}$*,* $\mathcal{A}_{\mathsf{Sig}}$*,* $\mathcal{A}_{\mathsf{sPKE}}$*, and* $\mathcal{A}_{\mathsf{rob}}$ *(which are all roughly as efficient as emulating an execution of* $\mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathsf{ACE},\mathcal{A}}$*) such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathsf{ACE},\mathcal{A}} \le\; & 2 \cdot \mathsf{Adv}^{\mathsf{PRF}}_{F,\mathcal{A}_{\mathsf{PRF}}} + 2 \cdot \mathsf{Adv}^{\mathsf{NIZK\text{-}ext}_1}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}_1}} + 4 \cdot \mathsf{Adv}^{\mathsf{NIZK\text{-}ext}_2}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}_2}} + 4 \cdot \mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}}} \\ & + (q_{S_1} + q_{R_1} + q_{D_1})^2 \cdot \mathsf{Adv}^{\mathsf{sPKE\text{-}SAN\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}} + 4(q_{R_1} + q_{R_2}) \cdot \mathsf{Adv}^{\mathsf{sPKE\text{-}USROB}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{rob}}}.\end{aligned}$$

Proof. Let $H_0 := \mathsf{Exp}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathsf{ACE},\mathcal{A}}$, let H_1 be as H_0 where F_K is replaced by a truly uniform random function U, and let H_2 be as H_1, where $crs^{\mathsf{NIZK}} \leftarrow \mathsf{NIZK.Gen}(1^\kappa)$ in $\mathsf{ACE.Setup}$ is replaced by $\big(crs^{\mathsf{NIZK}}, \xi^{\mathsf{NIZK}}\big) \leftarrow E_1^{\mathsf{NIZK}}(1^\kappa)$. Let W_{ACE} denote the event that $\mathcal{A}$ wins, i.e.,

$$W_{\mathsf{ACE}} := \big[b' = b \;\wedge\; c'_0 \neq \perp \neq c'_1 \;\wedge\; \forall j \in J\ m_{0,j} = m_{1,j} = \perp\big].$$

Similarly as in the proof of Lemma 6.3, it can be shown that there exist $\mathcal{A}_{\mathsf{PRF}}$ and $\mathcal{A}_{\mathsf{ZK}_1}$ such that

$$\Pr^{H_0}[W_{\mathsf{ACE}}] - \Pr^{H_2}[W_{\mathsf{ACE}}] = \mathsf{Adv}^{\mathsf{PRF}}_{F,\mathcal{A}_{\mathsf{PRF}}} + \mathsf{Adv}^{\mathsf{NIZK\text{-}ext}_1}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}_1}}. \tag{13}$$

Let H_3 be identical to H_2 except that after $\mathcal{A}_1$ returns $\big(c_0 = (\tilde{c}_0, \pi_0^{\mathsf{NIZK}}), c_1 = (\tilde{c}_1, \pi_1^{\mathsf{NIZK}}), st\big)$, H_3 executes for $\tilde{b} \in \{0,1\}$

$$w_{\tilde{b}} := \big(ek_{i_{\tilde{b}}}^{\mathsf{sPKE}}, m_{\tilde{b}}, r_{\tilde{b}}, vk_{i_{\tilde{b}}}^{\mathsf{Sig}}, \sigma_{i_{\tilde{b}}}^{\mathsf{Sig}}, \sigma_{c_{\tilde{b}}}^{\mathsf{Sig}}\big) \leftarrow E_2^{\mathsf{NIZK}}\big(crs^{\mathsf{NIZK}}, \xi^{\mathsf{NIZK}}, x_{\tilde{b}} := \big(vk^{\mathsf{Sig}}, \tilde{c}_{\tilde{b}}\big), \pi_{\tilde{b}}^{\mathsf{NIZK}}\big).$$

We clearly have

$$\Pr^{H_3}[W_{\mathsf{ACE}}] = \Pr^{H_2}[W_{\mathsf{ACE}}]. \tag{14}$$

Let $V_{\tilde{b}} := \big[\mathsf{NIZK.Ver}\big(crs^{\mathsf{NIZK}}, x_{\tilde{b}}, \pi_{\tilde{b}}^{\mathsf{NIZK}}\big) = 1\big]$ and let B_E be the event that (at least) one of the extractions fail, i.e.,

$$B_E := \big[(V_0 \ \wedge \ (x_0, w_0) \notin R) \ \vee \ (V_1 \ \wedge \ (x_1, w_1) \notin R)\big].$$

If B_E occurs, the knowledge extraction of NIZK is broken. To prove this, we define $\mathcal{A}_{\mathsf{ZK}_2}$ as follows. On input crs^{NIZK}, it emulates an execution of H_3, where in $\mathsf{ACE.Setup}$, crs^{NIZK} is used instead of generating it. When $\mathcal{A}_1$ returns (c_0, c_1, st), $\mathcal{A}_{\mathsf{ZK}_2}$ flips a coin $\tilde{b} \twoheadleftarrow \{0,1\}$ and returns $\big(x_{\tilde{b}}, \pi_{\tilde{b}}^{\mathsf{NIZK}}\big)$. If the $\tilde{b}$'s extraction fails, $\mathcal{A}_{\mathsf{ZK}_2}$ wins the extraction game. Hence,

$$\Pr^{H_3}[B_E] \leq 2 \cdot \mathsf{Adv}_{\mathsf{NIZK}, \mathcal{A}_{\mathsf{ZK}_2}}^{\mathsf{NIZK\text{-}ext}_2}. \tag{15}$$

For $\tilde{b} \in \{0,1\}$, let $B_{S,\tilde{b}}$ be the event that $(x_{\tilde{b}}, w_{\tilde{b}}) \in R$ and $ek_{i_{\tilde{b}}}^{\mathsf{sPKE}}$ is not contained in an answer from $\mathcal{O}_G$ to $\mathcal{A}_1$, and let B_S be the union of $B_{S,0}$ and $B_{S,1}$. We next show that if B_S occurs, the adversary found a forgery for the signature scheme.

Claim 1. *There exists a probabilistic algorithm $\mathcal{A}_{\mathsf{Sig}}$ such that*

$$\Pr^{H_3}[B_S] \leq 2 \cdot \mathsf{Adv}_{\mathsf{Sig}, \mathcal{A}_{\mathsf{Sig}}}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}. \tag{16}$$

Proof of claim. On input vk^{Sig}, $\mathcal{A}_{\mathsf{Sig}}$ emulate an execution of H_3, where vk^{Sig} is used in msk^{ACE} and sp^{ACE}. Queries $(i, \mathtt{sen})$ by $\mathcal{A}_1$ to the oracle $\mathcal{O}_G$ are answered by executing $\mathsf{ACE.Gen}$ (with F_K replaced by U) where σ_i^{Sig} is generated using the signing oracle of $\mathsf{Exp}_{\mathcal{E},\mathcal{A}}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}$. After extracting w_0 and w_1, $\mathcal{A}_{\mathsf{Sig}}$ flips a coin $\tilde{b} \twoheadleftarrow \{0,1\}$ and returns $\Big(\big[ek_{i_{\tilde{b}}}^{\mathsf{sPKE}}, vk_{i_{\tilde{b}}}^{\mathsf{Sig}}\big], \sigma_{i_{\tilde{b}}}^{\mathsf{Sig}}\Big)$. If $B_{S,\tilde{b}}$ occurs, $\big[ek_{i_{\tilde{b}}}^{\mathsf{sPKE}}, vk_{i_{\tilde{b}}}^{\mathsf{Sig}}\big]$ was not queried to the signing oracle and $(x_{\tilde{b}}, w_{\tilde{b}}) \in R$. The latter implies that $\sigma_{i_{\tilde{b}}}^{\mathsf{Sig}}$ is a valid signature and hence $\mathcal{A}_{\mathsf{Sig}}$ successfully forged a signature. We conclude

$$\Pr^{H_3}[B_S] \leq 2 \cdot \left(\frac{1}{2}\Pr^{H_3}[B_{S,0}] + \frac{1}{2}\Pr^{H_3}[B_{S,1}]\right) = 2 \cdot \mathsf{Adv}_{\mathsf{Sig}, \mathcal{A}_{\mathsf{Sig}}}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}. \qquad \Diamond$$

Let H_4 be identical to H_3 with the difference that we replace for $k \in \{0,1\}$ and $j \in J$, $m_{k,j} \leftarrow \mathsf{ACE.Dec}\big(\mathsf{ACE.Gen}(msk, j, \mathtt{rec}), c'_k\big)$ by

$$m_{k,j} \leftarrow \begin{cases} m_k, & ek_j^{\mathsf{sPKE}} = ek_{i_k}^{\mathsf{sPKE}} \text{ for } \big(ek_j^{\mathsf{sPKE}}, dk_j^{\mathsf{sPKE}}\big) = \mathsf{sPKE.Gen}\big(msk^{\mathsf{sPKE}}; U([j,0])\big), \\ \perp, & \text{else,} \end{cases} \tag{17}$$

where $ek_{i_k}^{\mathsf{sPKE}}$ are the extracted keys. Note that if V_k, $\neg B_E$, and $\neg B_S$ occur, we have $c'_k = \mathsf{San}(sp^{\mathsf{sPKE}}, \tilde{c}_k)$, $\tilde{c}_k = \mathsf{sPKE.Enc}\big(ek_{i_k}^{\mathsf{sPKE}}, m_k; r_k\big)$, and $ek_{i_k}^{\mathsf{sPKE}}$ was generated by $\mathcal{O}_G$. Hence, for $j \in J$ with $ek_j^{\mathsf{sPKE}} = ek_{i_k}^{\mathsf{sPKE}}$, we have by the correctness of the sPKE scheme that

$\mathsf{ACE.Dec}\big(\mathsf{ACE.Gen}(msk, j, \mathtt{rec}), c'_k\big) = m_k$, i.e., $m_{k,j} = m_k$ in both H_3 and H_4. For other $j \in J$, decryption only yields a message different from $\perp$ if robustness of the sPKE scheme is violated. Since $|J| \leq q_{R_1} + q_{R_2}$, this implies for $V \coloneqq V_0 \cap V_1$,

$$\Pr^{H_3}[W_{\mathsf{ACE}} \mid V \cap \neg B_E \cap \neg B_S] - \Pr^{H_4}[W_{\mathsf{ACE}} \mid V \cap \neg B_E \cap \neg B_S] \leq 2(q_{R_1}+q_{R_2})\mathsf{Adv}^{\mathsf{sPKE\text{-}USROB}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{rob}}}, \quad (18)$$

where $\mathcal{A}_{\mathsf{rob}}$ emulates the experiment and outputs $\tilde{c}_k$ for a uniformly chosen $k \in \{0,1\}$, i such that the i-th query to the key-generation oracle yields $ek_{i_k}^{\mathsf{sPKE}}$, and a uniformly chosen j.[6]

We finally construct an adversary $\mathcal{A}_{\mathsf{sPKE}}$ against the sanitization security of sPKE. On input $(sp^{\mathsf{sPKE}}, ek_0^{\mathsf{sPKE}}, ek_1^{\mathsf{sPKE}})$, $\mathcal{A}_{\mathsf{sPKE}}$ initializes $i_{q_0}, i_{q_1} \leftarrow \perp$, $k_q \leftarrow 1$, chooses distinct $q_0, q_1 \twoheadleftarrow \{1, \ldots, q_{S_1}+q_{R_1}+q_{D_1}\}$ uniformly at random, executes $\big(vk^{\mathsf{Sig}}, sk^{\mathsf{Sig}}\big) \leftarrow \mathsf{Sig.Gen}(1^\kappa)$, and $\big(crs^{\mathsf{NIZK}}, \xi^{\mathsf{NIZK}}\big) \leftarrow E_1^{\mathsf{NIZK}}(1^\kappa)$, and gives $sp^{\mathsf{ACE}} \coloneqq \big(sp^{\mathsf{sPKE}}, vk^{\mathsf{Sig}}, crs^{\mathsf{NIZK}}\big)$ to $\mathcal{A}_1$. It emulates the oracles for $\mathcal{A}_1$ as follows.

$\mathcal{O}_G(\cdot,\cdot)$**:** On query $(i, \mathtt{sen})$, if $k_q \notin \{q_0, q_1\}$ and $i \notin \{i_{q_0}, i_{q_1}\}$, generate an encryption key $\big(vk^{\mathsf{Sig}}, ek_i^{\mathsf{sPKE}}, sk_i^{\mathsf{Sig}}, \sigma_i^{\mathsf{Sig}}, crs^{\mathsf{NIZK}}\big)$ as H_4, where $\big(ek_i^{\mathsf{sPKE}}, dk_i^{\mathsf{sPKE}}\big)$ is obtained via $\mathcal{O}_G$ and remembered for future queries. If $k_q = q_l$ or $i = i_{q_l}$ for some $l \in \{0,1\}$, replace ek_i^{sPKE} by ek_l^{sPKE} and set $i_{q_l} \leftarrow i$. In both cases, set $k_q \leftarrow k_q + 1$ at the end.

On query $(j, \mathtt{rec})$, if $k_q \notin \{q_0, q_1\}$ and $j \notin \{i_{q_0}, i_{q_1}\}$, obtain a decryption key from $\mathcal{O}_G$, remember it, and set $k_q \leftarrow k_q + 1$. If $k_q = q_l$ or $j = i_{q_l}$ for some $l \in \{0,1\}$, then return $\perp$ and set $k_q \leftarrow k_q + 1$.

$\mathcal{O}_{SD}(\cdot,\cdot)$**:** On query $\big(j, c = \big(\tilde{c}, \pi^{\mathsf{NIZK}}\big)\big)$, if $k_q \notin \{q_0, q_1\}$ and $j \notin \{i_{q_0}, i_{q_1}\}$, then execute $c' \leftarrow \mathsf{ACE.San}(sp^{\mathsf{ACE}}, c)$, generate a decryption key dk_j^{ACE} as above, decrypt c' using dk_j^{ACE}, and return the resulting message. If $k_q = q_l$ or $j = i_{q_l}$ for some $l \in \{0,1\}$, set $i_{q_l} \leftarrow j$, if $\mathsf{NIZK.Ver}\big(crs^{\mathsf{NIZK}}, x \coloneqq \big(vk^{\mathsf{Sig}}, \tilde{c}, \big), \pi^{\mathsf{NIZK}}\big) = 0$, return $\perp$, otherwise, use the oracle $\mathcal{O}_{SD_l}$ of the sPKE-sanitization experiment to obtain a decryption of $\tilde{c}$ and return it. In all cases, set $k_q \leftarrow k_q + 1$ at the end.

When $\mathcal{A}_1$ returns $\big(c_0 = \big(\tilde{c}_0, \pi_0^{\mathsf{NIZK}}\big), c_1 = \big(\tilde{c}_1, \pi_1^{\mathsf{NIZK}}\big), st\big)$, $\mathcal{A}_{\mathsf{sPKE}}$ verifies the proofs π_0^{NIZK} and π_1^{NIZK} and extracts the witnesses to check the events V, B_E, and B_S. Denote by Q the event that $ek_{i_0}^{\mathsf{sPKE}}, ek_{i_1}^{\mathsf{sPKE}} \in \{ek_0^{\mathsf{sPKE}}, ek_1^{\mathsf{sPKE}}\}$, where $ek_{i_0}^{\mathsf{sPKE}}, ek_{i_1}^{\mathsf{sPKE}}$ are the extracted keys. Note that if V, $\neg B_E$, and $\neg B_S$ occur, both $ek_{i_0}^{\mathsf{sPKE}}$ and $ek_{i_1}^{\mathsf{sPKE}}$ have been returned by $\mathcal{O}_G$ to $\mathcal{A}_1$. This implies

$$\Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}SAN\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}[Q \mid V \cap \neg B_E \cap \neg B_S] \geq 1/(q_{S_1} + q_{R_1} + q_{D_1})^2. \quad (19)$$

If Q, V, $\neg B_E$, and $\neg B_S$ occur, $\mathcal{A}_{\mathsf{sPKE}}$ returns $(\tilde{c}_0, \tilde{c}_1)$ to the challenger of the sPKE-sanitization experiment to obtain the sanitized ciphertext $c'_{\tilde{b}}$. It then gives $\big(st, c'_{\tilde{b}}\big)$ to $\mathcal{A}_2$ and emulates the oracles as above. After $\mathcal{A}_2$ returned the bit b', $\mathcal{A}_{\mathsf{sPKE}}$ returns $b'' \leftarrow b'$. If $Q \cap V \cap \neg B_E \cap \neg B_S$ does not occur, $\mathcal{A}_{\mathsf{sPKE}}$ runs $\bar{c} \leftarrow \mathsf{sPKE.Enc}(ek_0^{\mathsf{sPKE}}, \bar{m})$ for an arbitrary fixed message $\bar{m}$ and returns $(c_0 \coloneqq \bar{c}, c_1 \coloneqq \bar{c})$ to the challenger. After receiving back a sanitized ciphertext $c'_{\tilde{b}}$, it returns a uniform bit $b'' \twoheadleftarrow \{0,1\}$.

Let W_{sPKE} be the event that $\mathcal{A}_{\mathsf{sPKE}}$ wins, i.e.,

$$W_{\mathsf{sPKE}} \coloneqq \big[b'' = \tilde{b} \ \wedge \ \exists j, j' \in \{0,1\}\ m_{0,j}^{\mathsf{sPKE}} \neq \perp \neq m_{1,j'}^{\mathsf{sPKE}})\big],$$

[6] Note that robustness is only defined for encryption and decryption keys generated by $\mathsf{sPKE.Gen}$. Hence, it is important to also condition on $\neg B_S$.

where the messages refer to the ones generated by $\mathsf{Exp}^{\mathsf{sPKE\text{-}SAN\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}$. Note that if $Q \cap V \cap \neg B_E \cap \neg B_S$ does not occur, we have $m^{\mathsf{sPKE}}_{0,0} = m^{\mathsf{sPKE}}_{1,0} = \bar{m} \neq \bot$ by the correctness of sPKE, and thus

$$\Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}SAN\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}[W_{\mathsf{sPKE}} \mid \neg(Q \cap V \cap \neg B_E \cap \neg B_S)] = \frac{1}{2}. \tag{20}$$

Next consider the case that $Q \cap V \cap \neg B_E \cap \neg B_S$ occurs. In this case, the view of $\mathcal{A}$ is identical to the one in H_4 with $b = \tilde{b}$, as long as the emulated $\mathcal{O}_G$ never returns $\bot$. Moreover, if $\mathcal{A}$ wins, we have $m^{H_4}_{0,j} = m^{H_4}_{1,j} = \bot$ for all $j \in J^{H_4}$, where the messages here refer to the ones in H_4, generated according to (17), and J^{H_4} is the set of all j such that $\mathcal{A}_1$ or $\mathcal{A}_2$ issued the query $(j, \mathtt{rec})$ to the oracle $\mathcal{O}_G$. Therefore, $\mathcal{O}_G$ is never gets a query for which it returns $\bot$ in this case. The event $Q \cap V \cap \neg B_E$ implies that the ciphertexts are encryptions of some message under ek^{sPKE}_0 or ek^{sPKE}_1. Correctness of sPKE now implies that $m^{\mathsf{sPKE}}_{0,0} \neq \bot \neq m^{\mathsf{sPKE}}_{1,0}$, i.e., the winning condition for $\mathcal{A}_{\mathsf{sPKE}}$ is satisfied. We can conclude that

$$\Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}SAN\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}[W_{\mathsf{sPKE}} \mid Q \cap V \cap \neg B_E \cap \neg B_S] \geq \Pr^{H_4}[W_{\mathsf{ACE}} \mid V \cap \neg B_E \cap \neg B_S]. \tag{21}$$

Let

$$p_G := \Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}SAN\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}[Q \cap V \cap \neg B_E \cap \neg B_S].$$

Putting our results together, we obtain

$$\begin{aligned}
\Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}SAN\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}[W_{\mathsf{sPKE}}] &= \Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}SAN\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}[W_{\mathsf{sPKE}} \mid Q \cap V \cap \neg B_E \cap \neg B_S] \cdot p_G \\
&\quad + \Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}SAN\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}[W_{\mathsf{sPKE}} \mid \neg(Q \cap V \cap \neg B_E \cap \neg B_S)] \cdot (1 - p_G) \\
&\overset{(20)}{=} \Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}SAN\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}[W_{\mathsf{sPKE}} \mid Q \cap V \cap \neg B_E \cap \neg B_S] \cdot p_G + \frac{1}{2}(1 - p_G).
\end{aligned}$$

This implies

$$\begin{aligned}
\Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}SAN\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}[W_{\mathsf{sPKE}} \mid Q \cap V \cap \neg B_E \cap \neg B_S] &= \frac{1}{p_G}\left[\Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}SAN\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}[W_{\mathsf{sPKE}}] - \frac{1}{2}(1 - p_G)\right] \\
&= \frac{1}{2\,p_G} \cdot \mathsf{Adv}^{\mathsf{sPKE\text{-}SAN\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}} + \frac{1}{2}.
\end{aligned} \tag{22}$$

Furthermore,

$$\mathsf{Adv}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathsf{ACE},\mathcal{A}} = 2 \cdot \Pr^{H_0}[W_{\mathsf{ACE}}] - 1 \overset{(13),(14)}{=} 2 \cdot \left(\mathsf{Adv}^{\mathsf{PRF}}_{F,\mathcal{A}_{\mathsf{PRF}}} + \mathsf{Adv}^{\mathsf{NIZK\text{-}ext}_1}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}_1}} + \Pr^{H_3}[W_{\mathsf{ACE}}]\right) - 1.$$

Since B_E, $\neg B_E \cap B_S$, and $\neg B_E \cap \neg B_S$ partition the sample space, the law of total probability implies

$$\begin{aligned}
\Pr^{H_3}[W_{\mathsf{ACE}}] &= \Pr^{H_3}[W_{\mathsf{ACE}} \cap B_E] + \Pr^{H_3}[W_{\mathsf{ACE}} \cap \neg B_E \cap B_S] \\
&\qquad + \Pr^{H_3}[W_{\mathsf{ACE}} \cap \neg B_E \cap \neg B_S] \\
&\leq \Pr^{H_3}[B_E] + \Pr^{H_3}[B_S] + \Pr^{H_3}[W_{\mathsf{ACE}} \cap \neg B_E \cap \neg B_S] \\
&\overset{(15),(16)}{\leq} 2 \cdot \mathsf{Adv}^{\mathsf{NIZK\text{-}ext}_2}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}_2}} + 2 \cdot \mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}}} + \Pr^{H_3}[W_{\mathsf{ACE}} \cap \neg B_E \cap \neg B_S].
\end{aligned}$$

Note that W_{ACE} implies $c'_0 \neq \bot \neq c'_1$ and thus also V because if the verification fails, $\mathsf{ACE.San}$ returns $\bot$. Hence,

$$\begin{aligned}
&\Pr^{H_3}[W_{\mathsf{ACE}} \cap \neg B_E \cap \neg B_S] = \Pr^{H_3}[W_{\mathsf{ACE}} \cap V \cap \neg B_E \cap \neg B_S] \\
&= \Pr^{H_3}[W_{\mathsf{ACE}} \mid V \cap \neg B_E \cap \neg B_S] \cdot \Pr^{H_3}[V \cap \neg B_E \cap \neg B_S] \\
&\overset{(18)}{\leq} \Big(\underbrace{\Pr^{H_4}[W_{\mathsf{ACE}} \mid V \cap \neg B_E \cap \neg B_S]} + 2(q_{R_1} + q_{R_2}) \cdot \mathsf{Adv}^{\mathsf{sPKE\text{-}USROB}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{rob}}}\Big) \cdot \Pr^{H_3}[V \cap \neg B_E \cap \neg B_S] \\
&\quad \overset{(21)}{\leq} \Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}SAN\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}[W_{\mathsf{sPKE}} | Q \cap V \cap \neg B_E \cap \neg B_S] \\
&\overset{(22)}{\leq} \left(\frac{1}{2\,p_G} \cdot \mathsf{Adv}^{\mathsf{sPKE\text{-}SAN\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}} + \frac{1}{2} + 2(q_{R_1} + q_{R_2}) \cdot \mathsf{Adv}^{\mathsf{sPKE\text{-}USROB}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{rob}}}\right) \cdot \Pr^{H_3}[V \cap \neg B_E \cap \neg B_S].
\end{aligned}$$

Since $\Pr^{H_3}[V \cap \neg B_E \cap \neg B_S] = \Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}SAN\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}[V \cap \neg B_E \cap \neg B_S]$, we have

$$\begin{aligned}
\frac{\Pr^{H_3}[V \cap \neg B_E \cap \neg B_S]}{p_G} &= \frac{\Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}SAN\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}[V \cap \neg B_E \cap \neg B_S]}{\Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}SAN\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}[Q \cap V \cap \neg B_E \cap \neg B_S]} \\
&= \left(\Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}SAN\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}[Q \mid V \cap \neg B_E \cap \neg B_S]\right)^{-1} \\
&\overset{(19)}{\leq} (q_{S_1} + q_{R_1} + q_{D_1})^2.
\end{aligned}$$

Therefore,

$$\Pr^{H_3}[W_{\mathsf{ACE}} \cap \neg B_E \cap \neg B_S] \leq \frac{1}{2}(q_{S_1} + q_{R_1} + q_{D_1})^2 \cdot \mathsf{Adv}^{\mathsf{sPKE\text{-}SAN\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}} + \frac{1}{2} + 2(q_{R_1} + q_{R_2}) \cdot \mathsf{Adv}^{\mathsf{sPKE\text{-}USROB}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{rob}}}.$$

This implies

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{ACE\text{-}SAN\text{-}CCA}}_{\mathsf{ACE},\mathcal{A}} \leq\ & 2 \cdot \mathsf{Adv}^{\mathsf{PRF}}_{F,\mathcal{A}_{\mathsf{PRF}}} + 2 \cdot \mathsf{Adv}^{\mathsf{NIZK\text{-}ext}_1}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}_1}} + 4 \cdot \mathsf{Adv}^{\mathsf{NIZK\text{-}ext}_2}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}_2}} + 4 \cdot \mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}}} \\
&+ (q_{S_1} + q_{R_1} + q_{D_1})^2 \cdot \mathsf{Adv}^{\mathsf{sPKE\text{-}SAN\text{-}CCA}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}} + 4(q_{R_1} + q_{R_2}) \cdot \mathsf{Adv}^{\mathsf{sPKE\text{-}USROB}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{rob}}}
\end{aligned}$$

and concludes the proof. □

We next prove non-detection of fresh encryptions, which directly follows from ciphertext unpredictability of the underlying sPKE scheme.

Lemma D.3. *Let* ACE *be the scheme from above and let* $\mathcal{A}$ *be an attacker on the non-detection of fresh encryptions that makes at most* q *queries to the oracle* $\mathcal{O}_G$*. Then, there exist probabilistic algorithms* $\mathcal{A}_{\mathsf{PRF}}$ *and* $\mathcal{A}_{\mathsf{sPKE}}$ *(which are both roughly as efficient as emulating an execution of* $\mathsf{Exp}^{\mathsf{ACE\text{-}NDTCT\text{-}FENC}}_{\mathsf{ACE},\mathcal{A}}$*) such that*

$$\mathsf{Adv}^{\mathsf{ACE\text{-}NDTCT\text{-}FENC}}_{\mathsf{ACE},\mathcal{A}} \leq \mathsf{Adv}^{\mathsf{PRF}}_{F,\mathcal{A}_{\mathsf{PRF}}} + (q+1) \cdot \mathsf{Adv}^{\mathsf{sPKE\text{-}UPD\text{-}CTXT}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}.$$

Proof. Let $H_0 \coloneqq \mathsf{Exp}^{\mathsf{ACE\text{-}NDTCT\text{-}FENC}}_{\mathsf{ACE},\mathcal{A}}$ and H_1 be as H_0 where F_K is replaced by a truly uniform random function U. As in the proof of Lemma 6.3, one can show that there exists $\mathcal{A}_{\mathsf{PRF}}$ such that

$$\Pr^{H_0}[b = 1] - \Pr^{H_1}[b = 1] = \mathsf{Adv}^{\mathsf{PRF}}_{F,\mathcal{A}_{\mathsf{PRF}}}.$$

The adversary $\mathcal{A}_{\mathsf{sPKE}}$ on input $\left(sp^{\mathsf{sPKE}}, ek^{\mathsf{sPKE}}, dk^{\mathsf{sPKE}}\right)$, sets $i_{q_0} \leftarrow \bot$, $k_q \leftarrow 1$, chooses $q_0 \twoheadleftarrow \{0, \ldots, q\}$ uniformly at random, runs $\left(vk^{\mathsf{Sig}}, sk^{\mathsf{Sig}}\right) \leftarrow \mathsf{Sig.Gen}(1^\kappa)$, $crs^{\mathsf{NIZK}} \leftarrow \mathsf{NIZK.Gen}(1^\kappa)$,

and gives $sp^{\mathsf{ACE}} := \big(sp^{\mathsf{sPKE}}, vk^{\mathsf{Sig}}, crs^{\mathsf{NIZK}}\big)$ to $\mathcal{A}$. It emulates the oracle $\mathcal{O}_G$ for $\mathcal{A}_1$ as follows. On query (i,t), if $k_q \neq q_0$ and $i \neq i_{q_0}$, then generate an encryption key $ek_i^{\mathsf{ACE}} := \big(vk^{\mathsf{Sig}}, ek_i^{\mathsf{sPKE}}, vk_i^{\mathsf{Sig}}, sk_i^{\mathsf{Sig}}, \sigma_i^{\mathsf{Sig}}, crs^{\mathsf{NIZK}}\big)$ and a decryption key $dk_i^{\mathsf{ACE}} := dk_i^{\mathsf{sPKE}}$ as H_1 does, where $\big(ek_i^{\mathsf{sPKE}}, dk_i^{\mathsf{sPKE}}\big)$ is obtained via $\mathcal{O}_G$ and remembered for future queries. Return ek_i^{ACE} if $t = \texttt{sen}$, and dk_i^{ACE} if $t = \texttt{rec}$. If $k_q = q_0$ or $i = i_{q_0}$, replace ek_i^{sPKE} and dk_i^{sPKE} by ek^{sPKE} and dk^{sPKE}, respectively, and set $i_{q_0} \leftarrow i$. In both cases, set $k_q \leftarrow k_q + 1$ at the end. When $\mathcal{A}$ returns $\big(m, i, c = (\tilde{c}, \pi^{\mathsf{NIZK}})\big)$, $\mathcal{A}_{\mathsf{sPKE}}$ returns $(m, \tilde{c})$.

Let Q be the event that $i_{q_0} = i$, or $q_0 = 0$ and $\mathcal{A}$ does not make the query $(i, \texttt{sen})$ or $(i, \texttt{rec})$ to $\mathcal{O}_G$. Note that the probability of Q is $1/(q+1)$ and since $b = \mathsf{ACE.DMod}\big(sp^{\mathsf{ACE}}, (\tilde{c}, \pi^{\mathsf{NIZK}}), (\tilde{c}^*, \pi^{\mathsf{NIZK}*})\big) = 1$ if and only if $\tilde{c}^* = \tilde{c}$, we have

$$\Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}UPD\text{-}CTXT}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}[c = c^* \mid Q] = \Pr^{H_1}[b = 1].$$

Hence, we can conclude

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{ACE\text{-}NDTCT\text{-}FENC}}_{\mathsf{ACE},\mathcal{A}} &= \Pr^{H_0}\big[b = 1\big] = \mathsf{Adv}^{\mathsf{PRF}}_{F,\mathcal{A}_{\mathsf{PRF}}} + \Pr^{H_1}[b = 1] \\
&= \mathsf{Adv}^{\mathsf{PRF}}_{F,\mathcal{A}_{\mathsf{PRF}}} + \Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}UPD\text{-}CTXT}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}[c = c^* \mid Q] \\
&\leq \mathsf{Adv}^{\mathsf{PRF}}_{F,\mathcal{A}_{\mathsf{PRF}}} + (q+1) \cdot \Pr^{\mathsf{Exp}^{\mathsf{sPKE\text{-}UPD\text{-}CTXT}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}}[c = c^*] \\
&= \mathsf{Adv}^{\mathsf{PRF}}_{F,\mathcal{A}_{\mathsf{PRF}}} + (q+1) \cdot \mathsf{Adv}^{\mathsf{sPKE\text{-}UPD\text{-}CTXT}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{sPKE}}}.
\end{aligned}$$

□

We finally prove the uniform decryption and role-respecting properties.

Lemma D.4. *Let* ACE *be the scheme from above and let* $\mathcal{A}$ *be an attacker on the uniform-decryption security that makes at most* q_R *queries of the form* $(\cdot, \texttt{rec})$ *to the oracle* $\mathcal{O}_G$*. Then, there exist probabilistic algorithms* $\mathcal{A}_{\mathsf{PRF}}$*,* $\mathcal{A}_{\mathsf{ZK}_1}$*,* $\mathcal{A}_{\mathsf{ZK}_2}$*,* $\mathcal{A}_{\mathsf{Sig}}$*, and* $\mathcal{A}_{\mathsf{rob}}$ *(which are all roughly as efficient as emulating an execution of* $\mathsf{Exp}^{\mathsf{ACE\text{-}URR}}_{\mathsf{ACE},\mathcal{A}}$*) such that*

$$\mathsf{Adv}^{\mathsf{ACE\text{-}UDEC}}_{\mathsf{ACE},\mathcal{A}} \leq \mathsf{Adv}^{\mathsf{PRF}}_{F,\mathcal{A}_{\mathsf{PRF}}} + \mathsf{Adv}^{\mathsf{NIZK\text{-}ext}_1}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}_1}} + \mathsf{Adv}^{\mathsf{NIZK\text{-}ext}_2}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}_2}} + \mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}}} + q_R \cdot \mathsf{Adv}^{\mathsf{sPKE\text{-}USROB}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{rob}}}.$$

Proof. Note that we can assume without loss of generality that $\mathcal{A}$ does not use the oracle $\mathcal{O}_E$ since obtaining encryption keys from $\mathcal{O}_G$ does not decrease the advantage. Let $H_0 := \mathsf{Exp}^{\mathsf{ACE\text{-}URR}}_{\mathsf{ACE},\mathcal{A}}$ and let W_{UDec} be the event that $\mathcal{A}$ wins the uniform-decryption game:

$$W_{\mathsf{UDec}} := \big[\exists j, j' \in J \; m_j \neq \bot \neq m_{j'} \;\wedge\; m_j \neq m_{j'}\big].$$

As in the proof of Lemma D.2, let H_1 be as H_0 with F_K replaced by a uniform random function U, let H_2 be as H_1 with crs^{NIZK} being generated by E_1^{NIZK}, and let H_3 be as H_2, but after $\mathcal{A}$ returns $c = \big(\tilde{c}, \pi^{\mathsf{NIZK}}\big)$, a witness

$$w := \big(ek_{i_w}^{\mathsf{sPKE}}, m_w, r_w, vk_{i_w}^{\mathsf{Sig}}, \sigma_{i_w}^{\mathsf{Sig}}, \sigma_{c,w}^{\mathsf{Sig}}\big)$$

for the statement $x := \big(vk^{\mathsf{Sig}}, \tilde{c}\big)$ is extracted from the proof π^{NIZK} by E_2^{NIZK}. We define the events $V := \big[\mathsf{NIZK.Ver}\big(crs^{\mathsf{NIZK}}, x, \pi^{\mathsf{NIZK}}\big) = 1\big]$, $B_E := \big[V \;\wedge\; (x,w) \notin R\big]$, and B_S as the event that $(x,w) \in R$ and $ek_{i_w}^{\mathsf{sPKE}}$ is not contained in an answer from $\mathcal{O}_G$ to $\mathcal{A}$. Is can be shown as in the proof of Lemma D.2 that there exist $\mathcal{A}_{\mathsf{PRF}}$, $\mathcal{A}_{\mathsf{ZK}_1}$, $\mathcal{A}_{\mathsf{ZK}_2}$, and $\mathcal{A}_{\mathsf{Sig}}$ such that

$$\begin{aligned}
\Pr^{H_0}[W_{\mathsf{UDec}}] - \Pr^{H_3}[W_{\mathsf{UDec}}] &= \mathsf{Adv}^{\mathsf{PRF}}_{F,\mathcal{A}_{\mathsf{PRF}}} + \mathsf{Adv}^{\mathsf{NIZK\text{-}ext}_1}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}_1}}, \\
\Pr^{H_3}[B_E] &\leq \mathsf{Adv}^{\mathsf{NIZK\text{-}ext}_2}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}_2}}, \\
\Pr^{H_3}[B_S] &\leq \mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}}},
\end{aligned}$$

where the last inequality uses that $\mathcal{A}$ does not query the oracle $\mathcal{O}_E$. Now let H_4 be as H_3 where for $j \in J$, $m_j \leftarrow \mathsf{ACE.Dec}\big(\mathsf{ACE.Gen}(msk, j, \mathtt{rec}), c'\big)$ is replaced by

$$m_j \leftarrow \begin{cases} m_w, & ek_j^{\mathsf{sPKE}} = ek_{i_w}^{\mathsf{sPKE}} \text{ for } \big(ek_j^{\mathsf{sPKE}}, dk_j^{\mathsf{sPKE}}\big) = \mathsf{sPKE.Gen}\big(msk^{\mathsf{sPKE}}; U([j,0])\big), \\ \bot, & \text{else.} \end{cases}$$

One can show as in the proof of Lemma D.2 that there exists a probabilistic algorithm $\mathcal{A}_{\mathsf{rob}}$ such that

$$\Pr^{H_3}[W_{\mathsf{UDec}} \mid V \cap \neg B_E \cap \neg B_S] - \Pr^{H_4}[W_{\mathsf{UDec}} \mid V \cap \neg B_E \cap \neg B_S] \leq q_R \cdot \mathsf{Adv}^{\mathsf{sPKE\text{-}USROB}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{rob}}}.$$

Note that $\mathcal{A}$ cannot win in H_4 since if $m_j \neq \bot \neq m_{j'}$, then $m_j = m_w = m_{j'}$. This implies that $\Pr^{H_3}[W_{\mathsf{UDec}} \mid V \cap \neg B_E \cap \neg B_S] \leq q_R \, \mathsf{Adv}^{\mathsf{sPKE\text{-}USROB}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{rob}}}$. Note that $\mathcal{A}$ can only win in H_3 if V occurs since otherwise $c' = \bot$ and consequently $m_j = \bot$ for all $j \in J$. We therefore obtain

$$\begin{aligned} \Pr^{H_3}[W_{\mathsf{UDec}}] &= \Pr^{H_3}[W_{\mathsf{UDec}} \cap V \cap B_E] + \Pr^{H_3}[W_{\mathsf{UDec}} \cap V \cap \neg B_E \cap B_S] \\ &\qquad + \Pr^{H_3}[W_{\mathsf{UDec}} \cap V \cap \neg B_E \cap \neg B_S] \\ &\leq \Pr^{H_3}[B_E] + \Pr^{H_3}[B_S] + \Pr^{H_3}[W_{\mathsf{UDec}} \mid V \cap \neg B_E \cap \neg B_S] \\ &\leq \mathsf{Adv}^{\mathsf{NIZK\text{-}ext}_2}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}_2}} + \mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}}} + q_R \cdot \mathsf{Adv}^{\mathsf{sPKE\text{-}USROB}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{rob}}}. \end{aligned}$$

Together with $\Pr^{H_0}[W_{\mathsf{UDec}}] - \Pr^{H_3}[W_{\mathsf{UDec}}] = \mathsf{Adv}^{\mathsf{PRF}}_{F,\mathcal{A}_{\mathsf{PRF}}} + \mathsf{Adv}^{\mathsf{NIZK\text{-}ext}_1}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}_1}}$, this concludes the proof. □

Lemma D.5. *Let* ACE *be the scheme from above and let $\mathcal{A}$ be an attacker on the role-respecting security that makes at most q_S queries of the form $(\cdot, \mathtt{sen})$ and at most q_R queries of the form $(\cdot, \mathtt{rec})$ to the oracle $\mathcal{O}_G$, and at most q_E queries to the oracle $\mathcal{O}_E$. Then, there exist probabilistic algorithms $\mathcal{A}_{\mathsf{PRF}}$, $\mathcal{A}_{\mathsf{ZK}_1}$, $\mathcal{A}_{\mathsf{ZK}_2}$, $\mathcal{A}_{\mathsf{Sig}}$, and $\mathcal{A}_{\mathsf{rob}}$ (which are all roughly as efficient as emulating an execution of $\mathsf{Exp}^{\mathsf{ACE\text{-}URR}}_{\mathsf{ACE},\mathcal{A}}$) such that*

$$\begin{aligned} \mathsf{Adv}^{\mathsf{ACE\text{-}RR}}_{\mathsf{ACE},\mathcal{A}} \leq \mathsf{Adv}^{\mathsf{PRF}}_{F,\mathcal{A}_{\mathsf{PRF}}} + \mathsf{Adv}^{\mathsf{NIZK\text{-}ext}_1}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}_1}} + \mathsf{Adv}^{\mathsf{NIZK\text{-}ext}_2}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}_2}} + (q_E+1) \cdot \mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}}} \\ + q_R \cdot \mathsf{Adv}^{\mathsf{sPKE\text{-}USROB}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{rob}}} + (q_S+q_R+q_E)^2 \cdot \mathsf{Col}^{\mathsf{ek}}_{\mathsf{sPKE}}. \end{aligned}$$

Proof. Let $H_0, \ldots, H_4$, $V \coloneqq \big[\mathsf{NIZK.Ver}\big(crs^{\mathsf{NIZK}}, x, \pi^{\mathsf{NIZK}}\big) = 1\big]$, and $B_E \coloneqq \big[V \wedge (x,w) \notin R\big]$ for the statement $x \coloneqq \big(vk^{\mathsf{Sig}}, \tilde{c}\big)$ and the extracted witness $w \coloneqq \big(ek_{i_w}^{\mathsf{sPKE}}, m_w, r_w, vk_{i_w}^{\mathsf{Sig}}, \sigma_{i_w}^{\mathsf{Sig}}, \sigma_{c,w}^{\mathsf{Sig}}\big)$ be defined as in the proof of Lemma D.4, and let W_{RR} be the event that $\mathcal{A}$ wins the role-respecting game:

$$W_{\mathsf{RR}} \coloneqq \big[c' \neq \bot \ \wedge\ \mathtt{dct} = \mathtt{false} \ \wedge\ \neg\big(\exists i \in I\ \forall j \in J\ (m_j \neq \bot \leftrightarrow P(i,j) = 1)\big)\big].$$

As in that proof, there exist $\mathcal{A}_{\mathsf{PRF}}$, $\mathcal{A}_{\mathsf{ZK}_1}$, and $\mathcal{A}_{\mathsf{ZK}_2}$ such that

$$\Pr^{H_0}[W_{\mathsf{RR}}] - \Pr^{H_3}[W_{\mathsf{RR}}] = \mathsf{Adv}^{\mathsf{PRF}}_{F,\mathcal{A}_{\mathsf{PRF}}} + \mathsf{Adv}^{\mathsf{NIZK\text{-}ext}_1}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}_1}}, \tag{23}$$

and

$$\Pr^{H_3}[B_E] \leq \mathsf{Adv}^{\mathsf{NIZK\text{-}ext}_2}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}_2}}. \tag{24}$$

Let E_G be the event that the extracted key $ek_{i_w}^{\mathsf{sPKE}}$ is contained in an answer from $\mathcal{O}_G$ to $\mathcal{A}$. One can show similarly as in the proof of Lemma D.2 that there exists an algorithm $\mathcal{A}_{\mathsf{rob}}$ such that

$$\Pr^{H_3}[W_{\mathsf{RR}} \cap V \cap \neg B_E \cap E_G] - \Pr^{H_4}[W_{\mathsf{RR}} \cap V \cap \neg B_E \cap E_G] \leq q_R \cdot \mathsf{Adv}^{\mathsf{sPKE\text{-}USROB}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{rob}}}. \tag{25}$$

We first show that if V, $\neg B_E$, and E_G occur in H_4, $\mathcal{A}$ can only win if two encryption keys generated by $\mathsf{sPKE.Gen}$ are equal, which happens only with negligible probability.

Claim 1. *We have*

$$\Pr^{H_4}[W_{\mathsf{RR}} \cap V \cap \neg B_E \cap E_G] \leq (q_S + q_R + q_E)^2 \cdot \mathsf{Col}^{\mathsf{ek}}_{\mathsf{sPKE}}.$$

Proof of claim. If V, $\neg B_E$, and E_G occur, then there is an $i_0 \in I$ such that $ek_{i_0}^{\mathsf{sPKE}} = ek_{i_w}^{\mathsf{sPKE}}$ for $\big(ek_{i_0}^{\mathsf{sPKE}}, dk_{i_0}^{\mathsf{sPKE}}\big) = \mathsf{sPKE.Gen}\big(msk^{\mathsf{sPKE}}; U([i_0, 0])\big)$. Using $P(i,j) = 1 \leftrightarrow i = j$, we have that $\mathcal{A}$ only wins if there exists $j \in J \backslash \{i_0\}$ such that $m_j \neq \bot$ or if $i_0 \in J$ and $m_{i_0} = \bot$. Because in H_4, m_j for $j \in J$ is equal to m_w if $ek_j^{\mathsf{sPKE}} = ek_{i_w}^{\mathsf{sPKE}}$ for $\big(ek_j^{\mathsf{sPKE}}, dk_j^{\mathsf{sPKE}}\big) = \mathsf{sPKE.Gen}\big(msk^{\mathsf{sPKE}}; U([j,0])\big)$, and $\bot$ otherwise, we have $m_{i_0} \neq \bot$ if $i_0 \in J$. Moreover, for $i_0 \neq j \in J$, we have $m_j = \bot$ unless $ek_j^{\mathsf{sPKE}} = ek_{i_0}^{\mathsf{sPKE}}$. This means that $\mathcal{A}$ can only win if $\mathsf{sPKE.Gen}$ generates the same encryption key for the randomness values $U([i_0, 0]$ and $U([j, 0]$ for some $i_0 \neq j \in J$. Since at most $q_S + q_R + q_E$ key pairs are generated in the experiment, there are at most $(q_S + q_R + q_E)^2$ pairs of encryption keys that could collide. For each such pair, the collision probability is bounded by $\mathsf{Col}^{\mathsf{ek}}_{\mathsf{sPKE}}$ because for $i \neq i'$, $U([i,0])$ and $U([i',0])$ are independent and uniformly distributed. Hence, the claim follows. ◊

Now let E_E be the event that $\mathcal{A}$ made a query $(i, \cdot)$ to $\mathcal{O}_E$ such that $ek_i^{\mathsf{sPKE}} = ek_{i_w}^{\mathsf{sPKE}}$ and $vk_i^{\mathsf{Sig}} = vk_{i_w}^{\mathsf{Sig}}$ for $\big(ek_i^{\mathsf{sPKE}}, dk_i^{\mathsf{sPKE}}\big) = \mathsf{sPKE.Gen}\big(msk^{\mathsf{sPKE}}; U([i,0])\big)$ and $\big(vk_i^{\mathsf{Sig}}, sk_i^{\mathsf{Sig}}\big) = \mathsf{Sig.Gen}\big(1^\kappa; U([i,1])\big)$. We next show that if $\mathcal{A}$ wins and $V \cap \neg B_E \cap \neg E_G \cap E_E$ occurs, $\mathcal{A}$ forged a signature on $\tilde{c}$.

Claim 2. *There exists a probabilistic algorithm $\mathcal{A}_{\mathsf{Sig}_1}$ such that*

$$\Pr^{H_3}[W_{\mathsf{RR}} \cap V \cap \neg B_E \cap \neg E_G \cap E_E] \leq q_E \cdot \mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}_1}}.$$

Proof of claim. On input vk^{Sig^*}, $\mathcal{A}_{\mathsf{Sig}_1}$ initializes $i_{q_0} \leftarrow \bot$, $k_q \leftarrow 1$, chooses $q_0 \twoheadleftarrow \{1, \ldots, q_E\}$ uniformly at random, generates $\big(sp^{\mathsf{sPKE}}, msk^{\mathsf{sPKE}}\big) \leftarrow \mathsf{sPKE.Setup}(1^\kappa)$, $\big(vk^{\mathsf{Sig}}, sk^{\mathsf{Sig}}\big) \leftarrow \mathsf{Sig.Gen}(1^\kappa)$, and $\big(crs^{\mathsf{NIZK}}, \xi^{\mathsf{NIZK}}\big) \leftarrow E_1^{\mathsf{NIZK}}(1^\kappa)$ as H_3, and gives $sp^{\mathsf{ACE}} := \big(sp^{\mathsf{sPKE}}, vk^{\mathsf{Sig}}, crs^{\mathsf{NIZK}}\big)$ to $\mathcal{A}$. It emulates the oracles for $\mathcal{A}$ as follows.

$\mathcal{O}_G(\cdot,\cdot)$**:** Generate the requested key exactly as H_3 does and return it.

$\mathcal{O}_E(\cdot,\cdot)$**:** On query $\big(i, m\big)$, if $k_q \neq q_0$ and $i \neq i_{q_0}$, generate an encryption key ek_i^{ACE} as H_3, encrypt m using ek_i^{ACE}, and return the resulting ciphertext. If $k_q = q_0$ or $i = i_{q_0}$, set $i_{q_0} \leftarrow i$, execute $\big(ek_i^{\mathsf{sPKE}}, dk_i^{\mathsf{sPKE}}\big) \leftarrow \mathsf{sPKE.Gen}\big(msk^{\mathsf{sPKE}}; U([i,0])\big)$, $\sigma_i^{\mathsf{Sig}} \leftarrow \mathsf{Sig.Sign}\big(sk^{\mathsf{Sig}}, \big[ek_i^{\mathsf{sPKE}}, vk_i^{\mathsf{Sig}}\big]; U([i,2])\big)$, and set $vk_i^{\mathsf{Sig}} := vk^{\mathsf{Sig}^*}$. Then, sample randomness r and compute $\tilde{c} \leftarrow \mathsf{sPKE.Enc}\big(ek_i^{\mathsf{sPKE}}, m; r\big)$, query the signing oracle on $\tilde{c}$ to obtain a signature σ_c^{Sig}, and run

$$\pi^{\mathsf{NIZK}} \leftarrow \mathsf{NIZK.Prove}\big(crs^{\mathsf{NIZK}}, x := \big(vk^{\mathsf{Sig}}, \tilde{c}\big), w := \big(ek_i^{\mathsf{sPKE}}, m, r, vk_i^{\mathsf{Sig}}, \sigma_i^{\mathsf{Sig}}, \sigma_c^{\mathsf{Sig}}\big)\big).$$

Finally, return the ciphertext $c := \big(\tilde{c}, \pi^{\mathsf{NIZK}}\big)$. In all cases, set $k_q \leftarrow k_q + 1$ at the end.

When $\mathcal{A}$ returns $c = \big(\tilde{c}, \pi^{\mathsf{NIZK}}\big)$, $\mathcal{A}_{\mathsf{Sig}_1}$ extracts a witness

$$w \coloneqq \big(ek_{i_w}^{\mathsf{sPKE}}, m_w, r_w, vk_{i_w}^{\mathsf{Sig}}, \sigma_{i_w}^{\mathsf{Sig}}, \sigma_{c,w}^{\mathsf{Sig}}\big) \leftarrow E_2^{\mathsf{NIZK}}\big(crs^{\mathsf{NIZK}}, \xi^{\mathsf{NIZK}}, x \coloneqq \big(vk^{\mathsf{Sig}}, \tilde{c}\big), \pi^{\mathsf{NIZK}}\big).$$

It finally returns the forgery attempt $\big(\tilde{c}, \sigma_{c,w}^{\mathsf{Sig}}\big)$.

Note that if $\mathcal{A}$ wins the role-respecting game, $\mathsf{ACE.DMod}\big(sp^{\mathsf{ACE}}, \hat{c}, c\big) = 0$ for all $\hat{c}$ that $\mathcal{O}_E$ has returned. Since $\mathsf{ACE.DMod}$ checks for equality of sPKE ciphertexts, this means that $\mathcal{A}_{\mathsf{Sig}_1}$ has not issued the query $\tilde{c}$ to its signing oracle. Furthermore, if the extraction and verification succeed, $\sigma_{c,w}^{\mathsf{Sig}}$ is a valid signature for $\tilde{c}$. Let Q be the event that $ek_{i_{q_0}}^{\mathsf{sPKE}} = ek_{i_w}^{\mathsf{sPKE}}$ and $vk_{i_{q_0}}^{\mathsf{Sig}} = vk_{i_w}^{\mathsf{Sig}}$. If Q and $V \cap \neg B_E \cap \neg E_G \cap E_E$ occur, $\mathcal{A}$ has not requested $ek_{i_{q_0}}^{\mathsf{ACE}}$ from $\mathcal{O}_G$ and hence $\mathcal{A}_{\mathsf{Sig}_1}$ perfectly emulates H_3. This implies

$$\Pr^{\mathsf{Exp}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}_1}}}[W_{\mathsf{Sig}} \mid V \cap \neg B_E \cap \neg E_G \cap E_E \cap Q] \geq \Pr^{H_3}[W_{\mathsf{RR}} \mid V \cap \neg B_E \cap \neg E_G \cap E_E],$$

where W_{Sig} denotes the event that $\mathcal{A}_{\mathsf{Sig}_1}$ wins in the signature forgery game. We further have

$$\Pr^{\mathsf{Exp}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}_1}}}[Q \mid V \cap \neg B_E \cap \neg E_G \cap E_E] = 1/q_E.$$

This implies for $p_G \coloneqq \Pr^{\mathsf{Exp}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}_1}}}[V \cap \neg B_E \cap \neg E_G \cap E_E \cap Q]$,

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}_1}} &= \Pr^{\mathsf{Exp}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}_1}}}[W_{\mathsf{Sig}}] \geq \Pr^{\mathsf{Exp}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}_1}}}[W_{\mathsf{Sig}} \mid V \cap \neg B_E \cap \neg E_G \cap E_E \cap Q] \cdot p_G \\
&\geq \Pr^{H_3}[W_{\mathsf{RR}} \mid V \cap \neg B_E \cap \neg E_G \cap E_E] \cdot p_G \\
&= \Pr^{H_3}[W_{\mathsf{RR}} \cap V \cap \neg B_E \cap \neg E_G \cap E_E] \cdot \frac{p_G}{\Pr^{H_3}[V \cap \neg B_E \cap \neg E_G \cap E_E]}.
\end{aligned}$$

Since $[V \cap \neg B_E \cap \neg E_G \cap E_E]$ in H_3 has the same probability as in $\mathsf{Exp}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}_1}}$, we have

$$\frac{p_G}{\Pr^{H_3}[V \cap \neg B_E \cap \neg E_G \cap E_E]} = \Pr^{\mathsf{Exp}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}_1}}}[Q \mid V \cap \neg B_E \cap \neg E_G \cap E_E] = \frac{1}{q_E},$$

which implies the claim. ◊

Finally, we show that if $\mathcal{A}$ wins and $V \cap \neg B_E \cap \neg E_G \cap \neg E_E$ occurs, $\mathcal{A}$ forged a signature on $\big[ek_{i_w}^{\mathsf{sPKE}}, vk_{i_w}^{\mathsf{Sig}}\big]$.

Claim 3. *There exists a probabilistic algorithm $\mathcal{A}_{\mathsf{Sig}_2}$ such that*

$$\Pr^{H_3}[W_{\mathsf{RR}} \cap V \cap \neg B_E \cap \neg E_G \cap \neg E_E] \leq \mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}_2}}.$$

Proof of claim. The algorithm $\mathcal{A}_{\mathsf{Sig}_2}$ on input vk^{Sig^*} runs $\big(sp^{\mathsf{sPKE}}, msk^{\mathsf{sPKE}}\big) \leftarrow \mathsf{sPKE.Setup}(1^\kappa)$ and $\big(crs^{\mathsf{NIZK}}, \xi^{\mathsf{NIZK}}\big) \leftarrow E_1^{\mathsf{NIZK}}(1^\kappa)$, and gives $sp^{\mathsf{ACE}} \coloneqq \big(sp^{\mathsf{sPKE}}, vk^{\mathsf{Sig}^*}, crs^{\mathsf{NIZK}}\big)$ to $\mathcal{A}$. It emulates the oracles for $\mathcal{A}$ as follows.

$\mathcal{O}_G(\cdot,\cdot)$**:** Generate the requested key as H_3, but obtain the signature σ_i^{Sig} via a query to the signing oracle. Remember the signature and when asked again for the same i, reuse σ_i^{Sig} instead of issuing another query. This ensures that the oracle behaves as the one in H_3 and returns the same key for repeated queries.

$\mathcal{O}_E(\cdot,\cdot)$: On query (i, m), generate an encryption key as for a query $(i, \texttt{sen})$ to $\mathcal{O}_G$, encrypt m using that key, and return the resulting ciphertext.

When $\mathcal{A}$ returns $c = (\tilde{c}, \pi^{\mathsf{NIZK}})$, $\mathcal{A}_{\mathsf{Sig}_1}$ extracts a witness

$$w := \left(ek_{i_w}^{\mathsf{sPKE}}, m_w, r_w, vk_{i_w}^{\mathsf{Sig}}, \sigma_{i_w}^{\mathsf{Sig}}, \sigma_{c,w}^{\mathsf{Sig}}\right) \leftarrow E_2^{\mathsf{NIZK}}\left(crs^{\mathsf{NIZK}}, \xi^{\mathsf{NIZK}}, x := \left(vk^{\mathsf{Sig}}, \tilde{c}\right), \pi^{\mathsf{NIZK}}\right).$$

It finally returns the forgery attempt $\left(\left[ek_{i_w}^{\mathsf{sPKE}}, vk_{i_w}^{\mathsf{Sig}}\right], \sigma_{i_w}^{\mathsf{Sig}}\right)$. Note that if $W_{\mathsf{RR}} \cap V \cap \neg B_E \cap \neg E_G \cap \neg E_E$ occurs, $\sigma_{i_w}^{\mathsf{Sig}}$ is a valid signature for $\left[ek_{i_w}^{\mathsf{sPKE}}, vk_{i_w}^{\mathsf{Sig}}\right]$ and $\mathcal{A}_{\mathsf{Sig}_2}$ has not requested a signature for this value from the signing oracle. Therefore, $\mathcal{A}_{\mathsf{Sig}_2}$ wins the forgery game and thus the probability of that event is bounded by $\mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}_2}}$. ◊

Combining Claims 2 and 3, we obtain

$$\Pr^{H_3}[W_{\mathsf{RR}} \cap V \cap \neg B_E \cap \neg E_G] \leq q_E \cdot \mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}_1}} + \mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}_2}}.$$

Let $\mathcal{A}_{\mathsf{Sig}}$ be the algorithm that runs $\mathcal{A}_{\mathsf{Sig}_1}$ with probability $\frac{q_E}{q_E+1}$ and $\mathcal{A}_{\mathsf{Sig}_2}$ with probability $\frac{1}{q_E+1}$. We then have

$$\begin{aligned}\mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}}} &= \frac{q_E}{q_E+1} \cdot \mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}_1}} + \frac{1}{q_E+1} \cdot \mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}_2}} \\ &\geq \frac{1}{q_E+1} \cdot \Pr^{H_3}[W_{\mathsf{RR}} \cap V \cap \neg B_E \cap \neg E_G].\end{aligned} \tag{26}$$

Note that W_{RR} implies $c' \neq \perp$ and therefore V, i.e., the events W_{RR} and $W_{\mathsf{RR}} \cap V$ are equal. Thus,

$$\begin{aligned}\Pr^{H_3}[W_{\mathsf{RR}}] &= \Pr^{H_3}[W_{\mathsf{RR}} \cap B_E] + \Pr^{H_3}[W_{\mathsf{RR}} \cap V \cap \neg B_E \cap E_G] + \Pr^{H_3}[W_{\mathsf{RR}} \cap V \cap \neg B_E \cap \neg E_G] \\ &\overset{(24),(25),(26)}{\leq} \mathsf{Adv}^{\mathsf{NIZK\text{-}ext}_2}_{\mathsf{NIZK},\mathcal{A}_{\mathsf{ZK}_2}} + q_R \cdot \mathsf{Adv}^{\mathsf{sPKE\text{-}USROB}}_{\mathsf{sPKE},\mathcal{A}_{\mathsf{rob}}} + \Pr^{H_4}[W_{\mathsf{RR}} \cap V \cap \neg B_E \cap E_G] \\ &\qquad + (q_E+1) \cdot \mathsf{Adv}^{\mathsf{Sig\text{-}EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{A}_{\mathsf{Sig}}}.\end{aligned}$$

Combined with Claim 1 and equation (23), this concludes the proof. □

References

[ABN10] M. Abdalla, M. Bellare, and G. Neven, "Robust encryption," in *Theory of Cryptography: 7th Theory of Cryptography Conference, TCC 2010*, D. Micciancio, Ed., Springer Berlin Heidelberg, 2010, pp. 480–497. DOI: 10.1007/978-3-642-11799-2_28.

[BL73] D. E. Bell and L. J. LaPadula, "Secure computer systems: Mathematical foundations," MITRE, Tech. Rep. MTR-2547, 1973.

[BBDP01] M. Bellare, A. Boldyreva, A. Desai, and D. Pointcheval, "Key-privacy in public-key encryption," in *Advances in Cryptology — ASIACRYPT 2001: 7th International Conference on the Theory and Application of Cryptology and Information Security*, C. Boyd, Ed., Springer Berlin Heidelberg, 2001, pp. 566–582. DOI: 10.1007/3-540-45682-1_33.

[BSW11] D. Boneh, A. Sahai, and B. Waters, "Functional encryption: Definitions and challenges," in *Theory of Cryptography: 8th Theory of Cryptography Conference, TCC 2011*, Y. Ishai, Ed., Springer Berlin Heidelberg, 2011, pp. 253–273. DOI: 10.1007/978-3-642-19571-6_16.

[CKN03] R. Canetti, H. Krawczyk, and J. B. Nielsen, "Relaxing chosen-ciphertext security," in *Advances in Cryptology - CRYPTO 2003: 23rd Annual International Cryptology Conference*, D. Boneh, Ed., Springer Berlin Heidelberg, 2003, pp. 565–582. DOI: 10.1007/978-3-540-45146-4_33.

[DHO16] I. Damgård, H. Haagh, and C. Orlandi, "Access control encryption: Enforcing information flow with cryptography," in *Theory of Cryptography: 14th International Conference, TCC 2016-B*, M. Hirt and A. Smith, Eds., Springer Berlin Heidelberg, 2016, pp. 547–576. DOI: 10.1007/978-3-662-53644-5_21.

[Elg85] T. Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, Jul. 1985. DOI: 10.1109/TIT.1985.1057074.

[FLPQ13] P. Farshim, B. Libert, K. G. Paterson, and E. A. Quaglia, "Robust encryption, revisited," in *Public-Key Cryptography – PKC 2013: 16th International Conference on Practice and Theory in Public-Key Cryptography*, K. Kurosawa and G. Hanaoka, Eds., Springer Berlin Heidelberg, 2013, pp. 352–368. DOI: 10.1007/978-3-642-36362-7_22.

[FGKO17] G. Fuchsbauer, R. Gay, L. Kowalczyk, and C. Orlandi, "Access control encryption for equality, comparison, and more," in *Public-Key Cryptography – PKC 2017: 20th IACR International Conference on Practice and Theory in Public-Key Cryptography*, S. Fehr, Ed., Springer Berlin Heidelberg, 2017, pp. 88–118. DOI: 10.1007/978-3-662-54388-7_4.

[GJJS04] P. Golle, M. Jakobsson, A. Juels, and P. Syverson, "Universal re-encryption for mixnets," in *Topics in Cryptology – CT-RSA 2004: The Cryptographers' Track at the RSA Conference 2004*, T. Okamoto, Ed., Springer Berlin Heidelberg, 2004, pp. 163–178. DOI: 10.1007/978-3-540-24660-2_14.

[GPSW06] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, ser. CCS '06, ACM, 2006, pp. 89–98. DOI: 10.1145/1180405.1180418.

[Gro04] J. Groth, "Rerandomizable and replayable adaptive chosen ciphertext attack secure cryptosystems," in *Theory of Cryptography: First Theory of Cryptography Conference, TCC 2004*, M. Naor, Ed., Springer Berlin Heidelberg, 2004, pp. 152–170. DOI: 10.1007/978-3-540-24638-1_9.

[Gro06] ——, "Simulation-sound NIZK proofs for a practical language and constant size group signatures," in *Advances in Cryptology – ASIACRYPT 2006: 12th International Conference on the Theory and Application of Cryptology and Information Security*, X. Lai and K. Chen, Eds., Springer Berlin Heidelberg, 2006, pp. 444–459. DOI: 10.1007/11935230_29.

[KW17] S. Kim and D. J. Wu, "Access control encryption for general policies from standard assumptions," in *Advances in Cryptology—ASIACRYPT 2017*, to appear, Springer Berlin Heidelberg, 2017.

[Lin06] Y. Lindell, "A simpler construction of CCA2-secure public-key encryption under general assumptions," *Journal of Cryptology*, vol. 19, no. 3, pp. 359–377, 2006. DOI: 10.1007/s00145-005-0345-x.

[NY90] M. Naor and M. Yung, "Public-key cryptosystems provably secure against chosen ciphertext attacks," in *Proceedings of the Twenty-second Annual ACM Symposium on Theory of Computing*, ser. STOC '90, ACM, 1990, pp. 427–437. DOI: 10.1145/100216.100273.

[PR07] M. Prabhakaran and M. Rosulek, "Rerandomizable RCCA encryption," in *Advances in Cryptology - CRYPTO 2007: 27th Annual International Cryptology Conference*, A. Menezes, Ed., Springer Berlin Heidelberg, 2007, pp. 517–534. DOI: 10.1007/978-3-540-74143-5_29.

[Sah99] A. Sahai, "Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security," in *40th Annual Symposium on Foundations of Computer Science*, 1999, pp. 543–553. DOI: 10.1109/SFFCS.1999.814628.

[SW05] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology – EUROCRYPT 2005: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, R. Cramer, Ed., Springer Berlin Heidelberg, 2005, pp. 457–473. DOI: 10.1007/11426639_27.

[TZMT17] G. Tan, R. Zhang, H. Ma, and Y. Tao, "Access control encryption based on LWE," in *Proceedings of the 4th ACM International Workshop on ASIA Public-Key Cryptography*, ser. APKC '17, ACM, 2017, pp. 43–50. DOI: 10.1145/3055504.3055509.