Journal of Universal Computer Science, vol. 25, no. 3 (2019), 158-181
submitted: 28/3/18, accepted: 30/12/18, appeared: 28/3/19 © J.UCS

Natural sd-RCCA Secure Public-key Encryptions from Hybrid Paradigms

Yuan Chen[1]
(State Key Laboratory of Integrated Services Networks, Xidian University
Xi'an, 710071, P.R. China
yuanchen@xidian.edu.cn)

Qingkuan Dong
(State Key Laboratory of Integrated Services Networks, Xidian University
Xi'an, 710071, P.R. China
qkdong@mail.xidian.edu.cn)

Yannan Li
(School of Computing and Information Technology, University of Wollongong
Wollongong, NSW 2522, Australia
yl738@uowmail.edu.au)

Qiqi Lai, Zhedong Wang
(Department of Computer and Electrical Engineering and Computer Science
Florida Atlantic University, Boca Raton, Florida, 33431, USA
qlai@fau.edu, wangz@fau.edu)

Abstract: The existence of natural public-key encryption (PKE) schemes satisfying secretly detectable replayable CCA (sd-RCCA) security is left as open. By introducing probabilistic message authentication codes (MACs) into popular KEM plus DEM paradigms, several instances of such schemes are presented in this paper. It is known that the encrypt-then-authenticate paradigm gives an RCCA secure DEM when the underlying MAC is regular (but not strong) secure, where forgeries for old messages might be possible. By further requiring that the validity of such forgeries can be verified only secretly, sd-RCCA secure DEMs is obtained. Combining such DEMs with CCA secure KEMs gives sd-RCCA secure hybrid PKEs. We first formalize the related notions and this paradigm, and also other variants of KEM plus DEM hybrid paradigm since MACs are commonly used in them. Then we show natural examples of desired probabilistic MACs under the standard DDH assumption, and find appropriate KEMs to match the message space for those MACs and then obtain natural instances of sd-RCCA secure hybrid PKEs.

Key Words: sd-RCCA security, probabilistic MAC, hybrid encryption, public-key encryption

Category: E.3, K.6.5

[1] Corresponding author.

1 Introduction

Replayable CCA (RCCA) security is a relaxed variant of CCA security for Public-Key Encryptions (PKE). It is proved to be sufficient for several cryptographic tasks [Canetti and Krawczky 2003, Coretti et al. 2013, Li et al. 2019, Maurer et al. 2012, Yu et al. 2017], and is believed to be sufficient for almost all the uses of CCA-secure encryptions [An et al. 2002, Shoup 2004]. In addition, it makes it possible to consider secure rerandomizable encryptions [Gröth 2004, Prabhakaran and Rosulek 2007].

In the definition of RCCA security, the decryption oracle answers '`test`' whenever a queried ciphertext decrypts to one of the questioned messages m_0 or m_1. This allows an adversary to modify a challenge ciphertext to another if the underlying plaintext is unchanged. Then by requiring such modification can be detected, RCCA security is strengthened. According to such detection can be done given only the public key or even the secret key, two stronger variants of RCCA security are introduced in [Canetti and Krawczky 2003], i.e, publicly detectable RCCA (pd-RCCA) and secretly detectable RCCA (sd-RCCA) security.

It is known that CCA ⇒ pd-RCCA ⇒ sd-RCCA ⇒ RCCA, and all the implications are strict. The two leftmost are shown in [Canetti and Krawczky 2003], and the rightmost is shown in [Prabhakaran and Rosulek 2007]. Nevertheless, almost all existing RCCA secure schemes satisfy the stronger pd-RCCA security, such as the schemes adding arbitrary padding to ciphertexts in the encryption while discarding it in the decryption, those allowing for more than one representation of ciphertexts, and even a recently proposed very natural LWE based schemes [El Bansarkhani et al. 2014]. "Natural" RCCA secure schemes satisfying only the weaker sd-RCCA security are left as an open problem in [Canetti and Krawczky 2003]. We will show such schemes in this paper. We simply denote sd-RCCA but not pd-RCCA security as sd-RCCA security later.

Now, let us first recall the two existing sd-RCCA secure constructions in [Canetti and Krawczky 2003], which are designed to show the gap between sd-RCCA and pd-RCCA security. The first one appends an encryption of m under an already sd-RCCA secure (possibly pd-RCCA secure since it implies sd-RCCA) scheme with an encryption of 0 under an independent pk. One can substitute the encryption of 0 by another, but the validity can be checked only secretly. Appending an encryption of 0 seems unnatural in practice.

Another one is related to rerandomizability. A PKE is rerandomizable if it is possible to convert an encryption c of m into another ciphertext c' that is indistinguishable from a fresh encryption of m. Depending on this can be done with just pk or still sk, the scheme could be publicly or secretly rerandomizable. Since a publicly rerandomizable RCCA scheme could be sd-RCCA secure but never pd-RCCA [Canetti and Krawczky 2003], this becomes a line for constructing sd-RCCA secure schemes. The second construction in [Canetti and Krawczky 2003] is given in this way. It applies an ElGamal encryption on an ciphertext under an already sd-RCCA secure (also possibly pd-RCCA secure) scheme to make it publicly rerandomizable. However, it seems difficult to build a natural scheme whose ciphertexts have only one group element to match the message space of ElGamal.

The two constructions in [Canetti and Krawczky 2003] are unnatural in practice. We follow another line in this paper, which simply follows the popular KEM+DEM hybrid paradigm [Cramer and Shoup 2003].

In the paradigm, KEM uses asymmetric techniques to encrypt a key, which is then used as the key by a symmetric cipher DEM to encrypt the message. It is well known that the combination of a CCA secure KEM with a (one-time) CCA secure DEM yield a CCA secue PKE. For RCCA security, similar result holds. For our purpose, we can relax one of the KEM and DEM to be sd-RCCA secure. In fact, sd-RCCA secure KEMs seem as difficult to be built as PKEs, so we seek for sd-RCCA secure DEMs.

We note that it has already been pointed out in [Canetti and Krawczky 2003] that an RCCA secure DEM can be achieved by combining a passive secure DEM with a regular (but not strong) secure message authentication code (MAC), since for a regular MAC it is possible to forge a new MAC value for an old message. Now, if for such MACs the validity of the forgery can be verified only secretly, then we obtain the desired DEMs. However, almost all practical MACs are deterministic, for which regular and strong security are equivalent, then we should find such natural MAC schemes from multi-value or probabilistic ones. Existing multi-value MACs are just conceptual or unnatural [Krawczyk 2001], so we turn to probabilistic MACs.

Probabilistic MACs have been recently proven to be useful and can be constructed efficiently from some standard hardness assumptions [Alwen et al. 2014, Dodis et al. 2012]. Some schemes in [Dodis et al. 2012] appear to meet our requirements. One may think that probabilistic MACs are overkilled since only one-time security for MACs is required, and information-theoretically secure ones exist. However, we are focus on such a stage that if the MAC in a CCA secure scheme is slightly weakened, then it might be naturally degenerated to an sd-RCCA secure one. This is the main reason why we deem our paradigm as "natural". Another reason is that when instantiating some hybrid encryptions with proper probabilistic MACs, we obtain sd-RCCA secure PKEs with very "natural" number theoretic operations as those in CCA secure ones. This mainly dues to the structures of these MACs. Also, if the efficiency is the problem, then our schemes are more efficient than the RCCA secure double-strand Cramer-shoup RCCA (rerandomizable) PKE [Prabhakaran and Rosulek 2007].

In section 2 and 3, we formalize the related notions and results mentioned above, there are some subtitles in the definition of sd-RCCA security for DEMs.

In section 4, we show two natural examples of MAC schemes as desired. The first one follows the construction from hash proof systems (HPS) in [Dodis et al. 2012], which is instantiated directly with a universal$_2$ HPS by Cramer and Shoup [Cramer and Shoup 2002], without the variant used in [Dodis et al. 2012]. The second one comes directly from [Dodis et al. 2012], which is the (so-called) full secure variant of the key-homomorphic weak PRF based construction when instantiated by a DDH-based example. From these MAC schemes, we further instantiate two natural PKE schemes as desired.

1.1 Further Discussions and Related Notions

Building RCCA secure schemes more efficient than CCA secure ones is another open problem left in [Canetti and Krawczky 2003]. Although a MAC scheme satisfying our requirement and more efficient than existing strong secure MAC schemes seems helpful, our schemes fail for that purpose. The reason is informally given in section 5.2.

Detectability is studied in isolation in [Hohenberger et al. 2012], where a notion called DCCA security is defined when danger can be detected publicly. Pd-RCCA is a natural case for DCCA security, but generally sd-RCCA is not. However, our schemes are obviously DCCA secure, thus show an overlap between sd-RCCA and DCCA security.

In our schemes, the underlying MACs are rerandomizable. However, the schemes as a whole are not fully rerandomizable. So, our paradigms do not help to provide RCCA secure rerandomizable PKEs.

2 Preliminaries

2.1 RCCA Security for PKE

Definition 1 (PKE). A public-key encryption (PKE) scheme consists of three algorithms. Probabilistic $\mathsf{PKE.Gen}$ that on input the security parameter k, generates public and private-keys (pk, sk), pk defines the message space $\mathcal{M}$. Probabilistic $\mathsf{PKE.Enc}$ encrypts a message $m \in \mathcal{M}$ into a ciphertext c by using pk. $\mathsf{PKE.Dec}$ decrypts cf by using sk, outputs either $m \in \mathcal{M}$ or a special symbol $\perp \notin \mathcal{M}$. Correctness is required, i.e, for all (pk, sk) generated by $\mathsf{PKE.Gen}$, and $m \in \mathcal{M}$, $\mathsf{PKE.Dec}_{sk}(\mathsf{PKE.Enc}_{pk}(m)) = m$.

Definition 2 (RCCA security for PKEs). We say a PKE scheme $\mathcal{PKE} = (\mathsf{PKE.Gen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ is RCCA secure if for every probablistic polynomial time oracle machine (PPT) $\mathcal{A}_E$ that plays the following game, its advantage $Adv^{\mathsf{rcca}}_{\Pi,\mathcal{A}_E}(k) = |Pr[\tilde{b} = b] - \frac{1}{2}|$ is negligible in k.

[RGAME.PKE]

Step 1. $(pk, sk) \leftarrow \mathsf{PKE.Gen}(1^k)$
Step 2. $(m_0, m_1, v) \leftarrow \mathcal{A}_E^{\mathcal{O}}(pk)$
Step 3. $b \leftarrow \{0,1\}$, $c \leftarrow \mathsf{PKE.Enc}_{pk}(m_b)$.
Step 4. $\tilde{b} \leftarrow \mathcal{A}_E^{\mathcal{O}}(v, c)$

By $\mathcal{O}$, we denote $\mathsf{PKE.Dec}_{sk}(\cdot)$, except that in step 4 $\mathcal{O}$ returns 'test' for any ciphertext decrypts to m_0 or m_1.

In RCCA secure schemes, a "replay" of plaintexts by modifying the ciphertext is allowed. Publicly-detectable (pd) and secretly-detectable (sd) RCCA security are defined according to whether the "replay" can be detected given pk or sk. The definitions are related to a notion of compatible relations. We now give the definitions in [Canetti and Krawczky 2003].

Definition 3 (Compatible relations for PKEs). For a PKE scheme $\mathcal{PKE} = (\mathsf{PKE.Gen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$, we say a family of binary relations $\equiv$ on ciphertext pairs is compatible, if for any (pk, sk) of $\mathcal{PKE}$, we have:

(a) For any ciphertexts c, c', if $c \equiv c'$, then $\mathsf{PKE.Dec}_{sk}(c) = \mathsf{PKE.Dec}_{sk}(c')$.

(b) For any $m \in \mathcal{M}$, if c and c' are two independent encryptions of m, then $\Pr[c \equiv c']$ is negligible in k.

Given c and c', if $\equiv$ can be computed efficiently with the sole knowledge of pk, then we say $\equiv$ is publicly computable, and rewrite it as $\equiv_{pk}$, if the computation needs also the knowledge of sk, then we say $\equiv$ is secretly computable, and rewrite it as $\equiv_{sk}$.

Definition 4 (pd-RCCA/sd-RCCA security for PKEs). We say $\mathcal{PKE}$ is pd-RCCA secure if there exists a publicly computable compatible relation $\equiv_{pk}$, such that $\mathcal{PKE}$ is secure according to the above definition of RCCA security with the modification that $\mathcal{O}$ returns test for any c' with $c' \equiv_{pk} c$. Denote the game as $\mathsf{pd\text{-}RGAME.PKE}$. We say $\mathcal{PKE}$ is sd-RCCA secure if the above holds for a secretly computable $\equiv_{sk}$. Denote the game as $\mathsf{sd\text{-}RGAME.PKE}$.

An observation In [Canetti and Krawczky 2003], it is pointed that (b) is redundant for pd-RCCA but necessary for sd-RCCA. However, we find that it is obscure to say that without (b), sd-RCCA security is trivially equivalent to RCCA. In fact, sd-RCCA security can never be achieved under compatible relations which do not satisfy (b).

For pd-RCCA, (b) is implied by CPA security, that is, if there is a publicly computable compatible relation $\equiv_{pk}$ such that $\mathcal{PKE}$ is pd-RCCA secure, then (b) must be satisfied by this $\equiv_{pk}$. This can be shown by constructing a CPA attacker $\mathcal{A}$ as follows: since there exists an m such that two independent encryptions satisfy $\equiv_{pk}$ with non-negligible probability in k, then $\mathcal{A}$ let $m_0 = m$ and randomly choose m_1, when the challenge ciphertext c^* is obtained, generate a random encryption c' of m_0, then check whether or not $c^* \equiv_{pk} c'$ publicly, if 'test' is returned, then return 0, else return 1. It's easy to see that the advantage of $\mathcal{A}$ is non-negligible.

For sd-RCCA, the compatible relation $\equiv_{sk}$ cannot be computed publicly, so the above attacker does not work. However, this time $\mathcal{A}$ can query c' to its decryption oracle, if 'test' is returned, then return 0, else return 1. It's easy to see that the advantage is non-negligible if (b) is not satisfied by $\equiv_{sk}$.

Therefore, if (b) is not required, then to achieve a meaningful notion, 'test' must also be returned for the encryptions of m_{1-b}, that is just what RCCA security requires.

It is shown in [Canetti and Krawczky 2003] that "CCA $\Rightarrow$ pd-RCCA $\Rightarrow$ sd-RCCA $\Rightarrow$ RCCA". We also noted that:

REMARK 1: For any pd/sd-RCCA secure scheme, $c \equiv c$ whatever $\equiv$ is, otherwise the scheme can never be pd/sd-RCCA secure.

REMARK 2: Since we are interested in RCCA secure schemes which are not CCA secure, the compatible relation $\equiv$ showing the pd- or sd-RCCA security must not be the equality relation: $c' \equiv c$ if $c' = c$. Therefore, we address that for such $\equiv$, publicly or secretly computable, it must be satisfied that it is easy to find a $c' \neq c$, such that $c' \equiv c$. That is, there exists a PPT machine, when given pk, c as inputs, it outputs a $c' \neq c$, such that $c' \equiv c$ with non-negligible probability. Otherwise, pd-RCCA or sd-RCCA secure schemes are trivially CCA secure.

2.2 KEM+DEM and related security notions

Definition 5 (KEM). A key encapsulation mechanism (KEM) consists of three algorithms. Probabilistic $\mathsf{KEM.Gen}$ that on input 1^k outputs a public/private key pair (pk, sk), pk defines the key space $\mathcal{K}_K$. Probabilistic encapsulation algorithm $\mathsf{KEM.Enc}$ that on input 1^k and a public key pk, outputs a pair (dk, ψ), where $dk \in \mathcal{K}_K$ is a key and ψ is its ciphertext. Decapsulation algorithm $\mathsf{KEM.Dec}$, on input sk and ψ, outputs either a key $dk \in \mathcal{K}_K$ or the special symbol $\perp$. Correctness is required, i.e, for all (pk, sk) generated by $\mathsf{KEM.Gen}$, and all $(dk, \psi) \leftarrow \mathsf{KEM.Enc}_{pk}(1^k)$, $\mathsf{KEM.Dec}_{sk}(\psi) = dk$.

Definition 6 (CCA for KEMs). We say a KEM scheme $\mathcal{KEM}$ =($\mathsf{KEM.Gen}$, $\mathsf{KEM.Enc}$, $\mathsf{KEM.Dec}$) is CCA secure if for every PPT $\mathcal{A}_K$ that plays the following game, its advantage $Adv^{\mathsf{cca}}_{\Pi,\mathcal{A}_K}(k) = |Pr[\tilde{\delta} = \delta] - \frac{1}{2}|$ is negligible in k.

[GAME.KEM]

Step 1. $(pk, sk) \leftarrow \mathsf{KEM.Gen}(1^k)$

Step 2. $(dk_1, \psi) \leftarrow \mathsf{KEM.Enc}_{pk}(1^k)$, $dk_0 \leftarrow \mathcal{K}_K$, $\delta \leftarrow \{0, 1\}$.

Step 3. $\tilde{\delta} \leftarrow \mathcal{A}_K^{\mathcal{O}}(pk, \psi, dk_\delta)$

$\mathcal{O}$ denotes $\mathsf{KEM.Dec}_{sk}(\cdot)$. In Step 3, $\mathcal{A}_K$ is restricted not to ask ψ to $\mathcal{O}$.

Definition 7 (DEM). A data encapsulation mechanism (DEM) is a one-time symmetric-key encryption, consists of two algorithms. $\mathsf{DEM.Enc}$ that takes as input 1^k, a key dk and a message $m \in \mathcal{M}$ ($\mathcal{M}$ is usually assumed to be $\{0,1\}^*$), outputs a ciphertext χ. $\mathsf{DEM.Dec}$ that takes as input a dk and a ciphertext χ, outputs a message m or the special symbol $\perp$. For our purpose, we allow $\mathsf{DEM.Enc}$ to be probabilistic. Correctness is required, i.e, for all $m \in \mathcal{M}$, $\mathsf{DEM.Dec}_{dk}(\mathsf{DEM.Enc}_{dk}(m)) = m$.

Definition 8 (OT/CCA/RCCA security for DEMs). We say that a DEM $\mathcal{DEM} = (\mathsf{DEM.Enc}\mathsf{DEM.Dec})$ is OT/CCA/RCCA secure, if for every PPT $\mathcal{A}_D$ that plays the following game, its advantage $Adv^{\mathsf{ot/cca/rcca}}_{\Pi,\mathcal{A}_D}(k) = |Pr[\tilde{b} = b] - \frac{1}{2}|$ is negligible in k.

[GAME.DEM]
Step 1. $(m_0, m_1, v) \leftarrow \mathcal{A}_D(1^k)$
Step 2. $dk \leftarrow \mathcal{K}_D$, $b \leftarrow \{0,1\}$, $\chi \leftarrow \mathsf{DEM.Enc}_{dk}(m_b)$.
Step 3. $\tilde{b} \leftarrow \mathcal{A}^{\mathcal{O}}_D(v, \chi)$

For the OT security, $\mathcal{O}$ is null. For the CCA security, $\mathcal{O}$ is $\mathsf{DEM.Dec}_{dk}(\cdot)$, and in Step 3 $\mathcal{A}_D$ is restricted not to ask χ to $\mathcal{O}$. For the RCCA security, all is the same except that in step 3 $\mathcal{O}$ returns 'test' for any ciphertext that decrypts to m_0 or m_1.

KEM+DEM hybrid paradigm works as follows, and it is well known that if $\mathcal{KEM}$ and $\mathcal{DEM}$ are IND-CCA secure then the following $\mathcal{HPKE}$ is IND-CCA secure (as a public-key encryption) [Cramer and Shoup 2003].

```
$\mathsf{HPKE.Enc}_{pk}(m)$
    $(dk, \psi) \leftarrow \mathsf{KEM.Enc}_{pk}()$
    $\chi \leftarrow \mathsf{DEM.Enc}_{dk}(m)$
    Output $c = (\psi, \chi)$
```

```
$\mathsf{HPKE.Dec}_{sk}(c)$
    $(\psi, \chi) \leftarrow c$
    $dk \leftarrow \mathsf{KEM.Dec}_{sk}(\psi)$
    $m \leftarrow \mathsf{DEM.Dec}_{dk}(\chi)$
    Output $m$
```

2.3 Detectable RCCA security for DEMs

To define pd-RCCA and sd-RCCA security for DEMs, we should first define compatible relations for them. We note that (b) in the definition of compatible relations for PKEs is not necessary now. Although our DEMs are randomized, it seems impossible for an adversary to generate random encryptions for both m_0 and m_1, since for DEMs we only require one-time security, so no encryption oracle is provided, thus the attack mentioned for PKEs doesn't work for DEMs. Due to this, we define compatible relations for DEMs without this requirement, which are simpler but sufficient for our purpose.

Definition 9 (compatible relations for DEMs). For a DEM scheme $\mathcal{DEM} = (\mathsf{DEM.Enc}, \mathsf{DEM.Dec})$, we say a family of binary relations $\equiv$ on ciphertext pairs is compatible, if for any dk of $\mathcal{DEM}$ and any ciphertexts c, c', if $c \equiv c'$, then $\mathsf{DEM.Dec}_{dk}(c) = \mathsf{DEM.Dec}_{dk}(c')$.

Given c and c', if $\equiv$ can be computed efficiently without the knowledge of dk, then we say $\equiv$ is publicly computable, if the computation needs the knowledge of dk, then we say $\equiv$ is secretly computable, and rewrite it as $\equiv_{dk}$.

Definition 10 (pd-RCCA/sd-RCCA for DEMs). We say that $\mathcal{DEM}$ is pd-RCCA secure if there exists a publicly computable compatible relation $\equiv$, such that $\mathcal{DEM}$ is secure according to the definition of RCCA security with the modification that $\mathcal{O}$ returns test for any c' with $c \equiv c'$. Denote the game as $\mathsf{pd\text{-}RGAME.DEM}$. We say $\mathcal{DEM}$ is sd-RCCA secure if the above holds for a secretly computable $\equiv_{dk}$. Denote the game as $\mathsf{sd\text{-}RGAME.DEM}$.

In fact, our definition of sd-RCCA security degrades to RCCA security. Nevertheless, it is sufficient for our purpose, and we still denote it as sd-RCCA to distinguish from pd-RCCA.

Theorem 11. *Let $\mathcal{DEM} = (\mathsf{DEM.Enc}, \mathsf{DEM.Dec})$ be an RCCA secure DEM scheme, then $\mathcal{DEM}$ is also sd-RCCA secure under the compatible relation $\equiv_{dk}$, where $\chi \equiv_{dk} \chi'$ if and only $\mathsf{DEM.Dec}_{dk}(\chi) = \mathsf{DEM.Dec}_{sk}(\chi')$.*

To prove this, we first show the following lemma.

Lemma 12. *Let $\mathcal{DEM} = (\mathsf{DEM.Enc}, \mathsf{DEM.Dec})$ be an RCCA secure DEM, then for any pair of messages $m_0, m_1 \in \mathcal{M}$, no efficient adversary can generate a random encryption of m_1 given that of m_0 with non-negligible security.*

Proof. If there exist a pair of messages $m_0, m_1 \in \mathcal{M}$, such that there is an efficient adversary $\mathcal{A}$, which can generate an encryption of m_1 given that of m_0, then we construct an RCCA attacker $\mathcal{B}$ as follows:

$\mathcal{B}$ lets $M_0 = m_0$, M_1 be a random $m \in \mathcal{M}$, then after receiving its challenge ciphertext χ^*, it runs $\mathcal{A}$ on χ^*. Let the result be χ', $\mathcal{B}$ submits χ' to its decryption oracle, if m_1 is returned then output 0, else output 1.

Now, Theorem 1 follows immediately.

Note that by requiring (b), a stronger notion of pd-RCCA/sd-RCCA for DEMs is obtained, and our reradomizable DEMs in Section 5 in fact show a gap between them.

3 Sd-RCCA secure hybrid public-key encryptions from sd-RCCA secure DEMs

It's easy to see that sd-RCCA secure KEMs lead to sd-RCCA secure hybrid encryptions, but such KEMs are almost as hard to be achieved as for PKEs. So, we seek for the other way. It can be proved that sd-RCCA security for DEMs is also sufficient.

Theorem 13. *If $\mathcal{KEM}$ is CCA-secure and $\mathcal{DEM}$ is sd-RCCA secure (but not pd-RCCA secure), then the hybrid scheme $\mathcal{HPKE}$ (as a PKE) by following KEM+DEM paradigm is sd-RCCA secure (but not pd-RCCA secure). In particular, for every $\mathcal{H}$, there exist $\mathcal{A}_K$ and $\mathcal{A}_D$ with*

$$Adv^{\mathrm{sd-rcca}}_{\mathcal{HPKE},\mathcal{H}}(k) \leq 2Adv^{\mathrm{cca}}_{\mathcal{KEM},\mathcal{A}_K}(k) + Adv^{\mathrm{sd-rcca}}_{\mathcal{DEM},\mathcal{A}_D}(k). \tag{1}$$

The proof follows the common game-modifying method, and we use the Shoup's Lemma [Cramer and Shoup 2003].

Lemma 14. *Let P, Q, and F be events defined on some probability space, such that* $\Pr[P \wedge \neg F] = \Pr[Q \wedge \neg F]$*, then* $|\Pr[P] - \Pr[Q]| \leq \Pr[F]$.

Proof Proof of Theorem 2. We first prove the sd-RCCA security of $\mathcal{HPKE}$ from the security of $\mathcal{KEM}$ and $\mathcal{DEM}$.

Let $\equiv_{dk}$ be a compatible relation for $\mathcal{DEM}$, we define a compatible relation for $\mathcal{HPKE}$ as follows: $(\psi, \chi) \equiv_{sk} (\psi', \chi')$ if $\psi = \psi'$ and $\chi \equiv_{dk} \chi'$ where $dk = \mathsf{KEM.Dec}_{sk}(\psi) = \mathsf{KEM.Dec}_{sk}(\psi')$. It is straightforward to verify $\equiv_{sk}$ is compatible for $\mathcal{HPKE}$ as long as $\equiv_{dk}$ is compatible for $\mathcal{DEM}$.

Now, let $\mathcal{H}$ be an adversary playing $\mathsf{sd\text{-}RGAME.PKE}$. Let (ψ^*, χ^*) be the challenge ciphertext, dk^* is the encapsulated key in ψ^*. We modify the game by using a random key dk^+ in place of dk^* in both the encryption and decryption oracle, i.e, dk^+ is used to form the challenge ciphertext, and a decryption oracle query is replied by using dk^+ whenever dk^* should be used. Call this game $\mathsf{sd\text{-}RGAME.PKE}'$. Let F and F' be events that $\tilde{b} = b$ in $\mathsf{sd\text{-}RGAME.PKE}$ and $\mathsf{sd\text{-}RGAME.PKE}'$, respectively. Then we claim that $|\Pr[F] - \Pr[F']| = 2Adv^{\mathrm{cca}}_{\mathcal{KEM}}(k)$, which is shown by constructing $\mathcal{A}_K$ that attacks the underlying KEM scheme by using $\mathcal{H}$.

$\mathcal{A}_K$ asks to obtain the challenge (pk, dk_δ, ψ^*) in $\mathsf{GAME.KEM}$, then sends pk to $\mathcal{H}$. After $\mathcal{H}$ chooses its m_0 and m_1, $\mathcal{A}_K$ randomly chooses $b \in \{0, 1\}$, computes $\chi^* = \mathsf{DEM.Enc}_{dk_\delta}(m_b)$, and sends (ψ^*, χ^*) to $\mathcal{H}$.

$\mathcal{A}_K$ answers $\mathcal{H}$'s decryption query (ψ, χ) as follows:

- If $\psi = \psi^*$ and so that $\chi \neq \chi^*$, then
  - If $\chi \equiv_{dk_\delta} \chi^*$ then $\mathcal{A}_K$ returns 'test' (note that $\mathcal{A}_K$ knows dk_δ).

- Else $\mathcal{A}_K$ uses dk_δ to decrypt χ, and returns the result to $\mathcal{H}$.

– If $\psi \neq \psi^*$, then $\mathcal{A}_K$ just forwards ψ to its own decryption oracle $\mathsf{KEM.Dec}_{sk}(\cdot)$.

- If $\perp$ is returned, then $\mathcal{A}_K$ returns $\perp$ to $\mathcal{H}$.
- If dk is returned, then $\mathcal{A}_K$ uses this dk to decrypt χ, and returns the result to $\mathcal{H}$.

This perfectly simulates the decryption oracle for $\mathcal{H}$. When $\mathcal{H}$ outputs $\tilde{b}$, $\mathcal{A}_K$ checks whether or not $\tilde{b} = b$, if so it outputs $\tilde{\delta} = 1$, else outputs $\tilde{\delta} = 0$. Now, we have $\Pr[\tilde{b} = b|\delta = 1] = \Pr[F]|$, and $\Pr[\tilde{b} = b|\delta = 0] = \Pr[F']$, then

$$\begin{aligned} Adv^{\rm cca}_{\mathcal{KEM},\mathcal{A}_K}(k) &= |\Pr[\tilde{\delta} = \delta] - \tfrac{1}{2}| = \tfrac{1}{2}|\Pr[\tilde{\delta} = 1|\delta = 1] - \Pr[\tilde{\delta} = 1|\delta = 0]| \\ &= \tfrac{1}{2}|\Pr[\tilde{b} = b|\delta = 1] - \Pr[\tilde{b} = b|\delta = 0]| = \tfrac{1}{2}|\Pr[F] - \Pr[F']| \end{aligned}$$

That is, $|\Pr[F] - \Pr[F']| = 2Adv^{\rm cca}_{\mathcal{KEM},\mathcal{A}_K}(k)$.

Next we argue that $\mathcal{H}$ in $\mathsf{sd\text{-}RGAME.PKE}'$ in fact conducts an attack against the sd-RCCA security of DEM, i.e. $|\Pr[F'] - \frac{1}{2}| = Adv^{\rm sd-rcca}_{\mathcal{DEM},\mathcal{A}_D}(k)$, where $\mathcal{A}_D$ is constructed as follows. $\mathcal{A}_D$ first runs $\mathsf{PKE.Gen}$ to generate (pk, sk), then sends pk to $\mathcal{H}$. After $\mathcal{H}$ chooses its (m_0, m_1), $\mathcal{A}_D$ gives them to its own encryption oracle and gets χ^*. Then $\mathcal{A}_D$ runs $\mathsf{KEM.Enc}$ to generate (dk^*, ψ^*), and gives (ψ^*, χ^*) to $\mathcal{H}$. It should be noticed that now the key dk^+ used in encryption oracle of $\mathsf{GAME.DEM}$ is chosen randomly from $\mathcal{K}_D$, so is independent of dk^*.

$\mathcal{A}_D$ answers $\mathcal{H}$'s decryption query (ψ, χ) as follows:

– If $\psi = \psi^*$ and so that $\chi \neq \chi^*$, then $\mathcal{A}_D$ forwards χ to its own decryption oracle, and returns the result to $\mathcal{H}$.

– If $\psi \neq \psi^*$, then $\mathcal{A}_D$ uses sk to decrypt ψ.

- If the result is $\perp$, then $\mathcal{A}_D$ returns $\perp$ to $\mathcal{H}$.
- If dk is returned, then $\mathcal{A}_D$ uses this dk to decrypt χ, and returns the result to $\mathcal{H}$.

When $\mathcal{H}$ outputs $\tilde{b}$, $\mathcal{A}_D$ outputs $\tilde{b}$, too. $\mathcal{A}_D$ perfectly simulates the game $\mathsf{sd\text{-}RGAME.PKE}'$, and $\mathcal{A}_D$ wins if $\mathcal{H}$ does. So, $|\Pr[F'] - \frac{1}{2}| = Adv^{\rm sd-rcca}_{\mathcal{DEM},\mathcal{A}_D}(k)$.

Finally, we have:

$$\begin{aligned} Adv^{\rm sd-rcca}_{\mathcal{HPKE},\mathcal{H}}(k) - Adv^{\rm sd-rcca}_{\mathcal{DEM},\mathcal{A}_D}(k) &= |\Pr[F] - \frac{1}{2}| - |\Pr[F'] - \frac{1}{2}| \\ &\leq |\Pr[F] - \Pr[F']| \\ &= 2Adv^{\rm cca}_{\mathcal{KEM},\mathcal{A}_K}(k). \end{aligned}$$

Then (1) follows immediately.

The major factors of the running time of $\mathcal{A}_D$ and $\mathcal{A}_K$ is that of $\mathcal{H}$ and that for simulating the decryption oracle which grow linearly in the number of decryption queries.

It remains to show $\mathcal{HPKE}$ is not pd-RCCA secure.

Since $\mathcal{DEM}$ is sd–RCCA secure but not pd-RCCA secure, there exists a compatible relation showing the sd-RCCA security, which is secretly but not publicly computable, let $\equiv_{dk}$ be the compatible relation. We claim that if χ satisfies $\chi \equiv_{dk} \chi^*$, then for any publicly computable compatible relation $\equiv_{pk}$, we must have $(\psi^*, \chi) \not\equiv_{pk} (\psi^*, \chi^*)$. If this is admitted, then the decryption of (ψ^*, χ) is m_b and the decryption oracle will not return 'test' , so $\mathcal{HPKE}$ is not pd-RCCA secure.

We now prove our claim. Intuitively, $\chi \equiv_{dk} \chi^*$ cannot be publicly computed, but if when given ψ^* this $\equiv_{dk}$ can be publicly computed, then ψ^* must reveal the information of dk^*, which contradicts with the CCA security of $\mathcal{KEM}$.

More formally, if there exist some publicly computable $\equiv_{pk}$, such that (ψ^*, χ) $\equiv_{pk}$ (ψ^*, χ^*), then we construct a CCA adversary $\mathcal{A}$ against $\mathcal{KEM}$ as follows: given (pk, ψ^*, dk_δ), $\mathcal{A}$ uses dk_δ to generate two DEM ciphertexts χ and χ' with $\chi \equiv_{dk} \chi'$, then checks whether or not $(\psi^*, \chi) \equiv_{pk} (\psi^*, \chi')$, if so output 1, else output 0.

Since $\equiv_{dk}$ cannot be publicly computed, if ψ^* encapsulates a dk independent of dk_δ, then except for a negligible probability, we have $(\psi^*, \chi) \not\equiv_{pk} (\psi^*, \chi^*)$. (Else $\chi \equiv_{dk} \chi^*$ can be publicly computed by randomly generating a ψ^* first, then publicly check whether or not $(\psi^*, \chi) \equiv_{pk} (\psi^*, \chi^*)$.)

Then it is easy to see $\Pr[\mathcal{A} = \delta]$ is almost 1.

4 sd-RCCA secure DEMs from regular secure and secretly detectable MACs

It has already been pointed out that RCCA secure SKEs can be given by the "encrypt-then-authenticate" paradigm by using a regular but not necessarily strong secure MAC. For sd-RCCA secure DEMs, we follow the same paradigm. However, the underlying MAC needs to be regular secure (but not strong one-time secure), and the validity of a successful forge can be verified only secretly (but not publicly). We now formalize these notions for MACs.

4.1 MAC and related security notions

Definition 15 (MAC). MAC is a pair of algorithms ($\mathsf{MAC.Sign}$, $\mathsf{MAC.Ver}$). A key space $\mathcal{K}_M$ is defined by security parameter k. $\mathsf{MAC.Sign}$ takes a key $mk \in \mathcal{K}_M$ and a message $m \in 0, 1^*$ as inputs, and outputs a string σ. $\mathsf{MAC.Ver}$ takes a triple (mk, m, σ) as input and outputs a decision of whether or not (m, σ) is valid with respect to mk.

If $\mathsf{MAC.Sign}$ is deterministic, then $\mathsf{MAC.Ver}$ can be done just by checking if $\sigma = MAC.Sign_{mk}(m)$. However, since we will use randomized MACs to achieve sd-RCCA secure DEMs, $\mathsf{MAC.Sign}$ is allowed to be probabilistic, but $\mathsf{MAC.Ver}$ is still deterministic.

For probabilistic MACs, a proper security notion should allow an adversary to make $\mathsf{MAC.Sign}(mk, \cdot)$ and $\mathsf{MAC.Ver}(mk, \cdot)$ queries [Bellare et al. 2004]. However, for our setting we need only one-time security, i.e, only once access to MAC sign is permitted. In fact, the weaker notion without access to $\mathsf{MAC.Ver}(mk, \cdot)$ is sufficient.

Definition 16 (regular/strong security for MACs). We say that a MAC scheme $\mathcal{MAC} = (\mathsf{MAC.Sign}, \mathsf{MAC.Ver})$ is secure against one-time chosen message attack, or shorten as regular one-time secure, if for every PPT oracle machine $\mathcal{F}$ that plays the following game, the probability that the game output 1 (i.e, the advantage of $\mathcal{F}$, denoted as $Adv^{\text{forge}}_{\mathcal{MAC},\mathcal{F}}(k)$) is negligible in k.

[GAME.MAC].

Step1. $m \leftarrow \mathcal{F}(1^k)$

Step2. $mk \leftarrow \mathcal{K}_M$, $\sigma \leftarrow \mathsf{MAC.Sign}_{mk}(m)$

Step3. $(m', \sigma') \leftarrow \mathcal{F}(\sigma)$

Step4. If $m' \neq m$ and $\mathsf{MAC.Ver}_{mk}(m', \sigma') = 1$ then output 1 else output 0

Strong one time security is defined all the same except that $m' \neq m$ is replaced with $(m', \sigma') \neq (m, \sigma)$ in step 4.

For deterministic MACs, the two definitions are equivalent. However, for a regular randomized MACs, it might be possible to efficiently generate another valid MAC value σ' for m, which is not allowed for a strong secure one. For such forgery, we distinguish two cases:

Definition 17 (Publicly/secretly detectable forgery). Let $\mathcal{MAC}$ be a regular secure (but not strong one-time secure) MAC and (m', σ') be a forgery output by an adversary when given (m, σ) with $m' = m$. Then we say $\mathcal{MAC}$ is publicly-detectable if given (m, σ, σ'), the validity of σ' can be verified efficiently without the knowledge of mk, else we say $\mathcal{MAC}$ is secretly-detectable.

4.2 Sd-RCCA secure DEMs from the "encrypt-then-authenticate" paradigm

One can obtain an sd-RCCA secure DEM easily by following the "encrypt-then-authenticate" paradigm from a regular MAC and a one-time secure DEM, and it is well known that the latter can be just a one-time pad. We now formalize the paradigm.

Theorem 18. *Let $\mathcal{DEM}^{ot}$ be a one-time secure (deterministic) DEM, $\mathcal{MAC}$ be a MAC which is regular secure (but not strong one-time secure), and is secretly-detectable (but not publicly-detectable), then the following DEM $\mathcal{DEM}^{sd-rcca}$ is sd-RCCA secure (but not pd-RCCA secure). In particular, the secretly computable compatible relation $\equiv_{dk,mk}$ should be $\chi = (c, \sigma) \equiv_{dk,mk} \chi' = (c', \sigma')$ if and only if $c = c'$, $\sigma \neq \sigma'$, and both $\mathsf{MAC.Ver}_{mk}(c, \sigma) = 1$ and $\mathsf{MAC.Ver}_{mk}(c', \sigma')$ $= 1$.*

```
$\mathsf{DEM.Enc}_{dk,mk}(m)$
  $c \leftarrow \mathsf{DEM.Enc}_{dk}(m)$
  $\sigma \leftarrow \mathsf{MAC.Sign}_{mk}(c)$
  Output $\chi = (c\|\sigma)$
```

```
$\mathsf{DEM.Dec}_{dk,mk}(\chi)$
  parse $\chi$ as $c\|\sigma$
  If $\mathsf{MAC.Ver}_{mk}(c, \sigma) = 1$ then
  $m \leftarrow \mathsf{DEM.Dec}_{dk}(c)$
  Else output $\perp$ EndIf
  Output $m$.
```

Proof. The compatibility of $\equiv_e$ is obvious. We first prove the sd-RCCA security.

Let $\mathcal{A}_D$ be an adversary playing $\mathsf{sd\text{-}RGAME.DEM}$, we construct a passive adversary $\mathcal{B}$ against $\mathcal{DEM}^{ot}$ by using $\mathcal{A}_D$ as follows:

$\mathcal{B}$ forwards 1^k to $\mathcal{A}_D$. Given (m_0, m_1) from $\mathcal{A}_D$, $\mathcal{B}$ requests (m_0, m_1) to the encryption oracle of $\mathsf{GAME.DEM}$ to obtain c^*. Then $\mathcal{B}$ randomly chooses mk from $\mathcal{K}_M$, computes $\sigma^* = MAC.Sign_{mk}(c^*)$, sends $\chi^* = (c^*, \sigma^*)$ to $\mathcal{A}_D$.

For a decryption query $\chi = (c, \sigma)$ from $\mathcal{A}_D$, if $c = c^*$, then $\mathcal{B}$ checks if $\mathsf{MAC.Ver}_{mk}(c^*, \sigma) = 1$ by using mk, if so, it returns 'test', for all other cases $\mathcal{B}$ just returns $\perp$.

Finally, when $\mathcal{A}_D$ outputs $\tilde{b}$, $\mathcal{B}$ outputs $\tilde{b}$, too.

The simulation is correct unless $\mathsf{MAC.Ver}(c, \sigma) = 1$ for some $c \neq c^*$. Let Forge denote this event, we have $\Pr[\mathsf{Forge}] \leq q_D \cdot Adv^{\mathsf{forge}}_{\mathcal{MAC},\mathcal{A}_D}$.

It remains to show $\mathcal{DEM}^{sd-rcca}$ is not pd-RCCA secure. Assume that there exist a publicly computable relation such that $\mathcal{DEM}$ is pd-RCCA secure, let $\equiv$ be the relation. Since the underlying $\mathcal{MAC}$ is secretly but not publicly detectable, it is possible to forge a new and valid σ' efficiently for c^*, but the validity of (c^*, σ') cannot be verified publicly. However, we note that it must be the case that $(c^*, \sigma') \equiv (c^*, \sigma^*)$, else the decryption of (c^*, σ') is m_b, thus $\mathcal{DEM}^{sd-rcca}$ cannot be pd-RCCA secure for this $\equiv$. Since $\equiv$ is a publicly computable relation, this means that the validity of σ' can be verified publicly, which leads to a contradict.

5 Achieving sd-RCCA security from regular MACs by other paradigms

There are also some other methods using MACs to achieve CCA secure hybrid encryptions, such as a CCCA secure KEM plus an authenticated encryption (which is shortened as AE and can be built from a passively secure DEM and a MAC) [Hofheinz and Kiltz 2007], a CCA secure Tag-KEM (which can be constructed by a LCCA secure KEM and a MAC) plus a passively secure DEM [Abe et al. 2008], an RCCA secure KEM plus a CCA secure Tag-DEM (which can be constructed by an OT secure DEM and a MAC)[Chen and Dong 2014], and so on. Instantiating the MAC underlying these constructions with a regular one-time secure (but not strong one-time secure), secretly-detectable (but not publicly-detectable) one will also yield sd-RCCA secure hybrid encryptions.

In section 5, we also instantiate a scheme for the CCCA secure KEM plus AE paradigm, so we formalized the paradigm here, the formal definition for the CCCA security for KEMs follows directly from [Hofheinz and Kiltz 2007] and is given in Appendix A.

Definition 19 (AE). An authenticated encryption (AE) scheme is a one-time symmetric-key encryption, consists of two algorithms. $\mathsf{AE.Enc}$ that takes as input 1^k, a key dk and a message $m \in \mathcal{M}$, outputs a ciphertext χ. $\mathsf{AE.Dec}$ that takes as input a dk and a ciphertext χ, outputs a message m or the special symbol $\perp$. For our purpose, we allow $\mathsf{AE.Enc}$ to be probabilistic. Correctness is required, i.e, for all $m \in \mathcal{M}$, $\mathsf{AE.Dec}_{dk}(\mathsf{AE.Enc}_{dk}(m)) = m$.

Definition 20 (OT/ROT security for AE). The one-time(OT) security of AE captures privacy and authenticity simultaneously, which is defined by the following game, where $\mathcal{O}$ is a decrypt-or-reject oracle, which returns $\mathsf{AE.Dec}_{dk}(\chi)$ if $b = 1$, else always returns $\perp$. In Step 3, $\mathcal{A}_A$ is allowed only one query to $\mathcal{O}$, which is restricted not to be χ.

[GAME.AE]
Step 1. $(m_0, m_1, v) \leftarrow \mathcal{A}_A(1^k)$
Step 2. $dk \leftarrow \mathcal{K}_D$, $b \leftarrow \{0, 1\}$, $\chi \leftarrow \mathsf{AE.Enc}_{dk}(m_b)$.
Step 3. $\tilde{b} \leftarrow \mathcal{A}_A^{\mathcal{O}}(v, \chi)$

Replayable one-time security (ROT) for AEs is defined similarly except that in step 3 $\mathcal{O}$ returns 'test' for any ciphertext that decrypts to m_0 or m_1, regardless of $b$and we make no restriction on the number of such queries.

The compatible relations for AEs are defined almost the same as for DEMs, then pd-ROT and sd-ROT security for AEs follow immediately, also with no restriction on the number of queries when 'test' is returned.

Remark In the definition of ROT, restricting once access to $\mathcal{O}$ when the ciphertexts decrypts to m_0 or m_1 is not reasonable, since this means even one more such replaying ciphertext query might be dangerous. However, we have mentioned that to make replayable style security meaningful, it should be easy to generate (more than one in our case) such replaying ciphertexts.

Theorem 21. *If $\mathcal{KEM}$ is CCCA-secure and $\mathcal{AE}$ is sd- but not pd-ROT secure, then the hybrid scheme $\mathcal{HPKE}$ by following KEM+DEM paradigm with the DEM substituted by an AE is sd- but not pd-RCCA secure(as a PKE).*

The proof for sd-RCCA security is similar as in [Hofheinz and Kiltz 2007], and the proof for not pd-RCCA security is almost the same as for Theorem 2. We show the details in Appendix A.

For sd- but not pd-ROT secure AEs, we can still follow the "encrypt-then-authenticate" paradigm.

Theorem 22. *Let $\mathcal{DEM}^{ot}$ be a one-time secure (deterministic) DEM, $\mathcal{MAC}$ be a MAC which is regular (but not strong) one-time secure, and is secretly-detectable (but not publicly-detectable), then the AE defined the same as in Theorem 3 is sd-RCCA secure (but not pd-RCCA secure).*

The proof is also almost the same as for Theorem 4.1, so we omit it here.

6 Instantiations

6.1 Instantiations of regular but not strong, secretly but not publicly detectable MACs

There are motivations for probabilistic MACs as pointed in [Dodis et al. 2012]. And such MACs give rise to natural regular but not strong MACs. For example, the constructions from labeled hash proof systems (HPS) when instantiate it directly with the universal$_2$ HPS by Cramer and Shoup [Cramer and Shoup 2002], the DDH-based constructions achieving full security from key homomorphic weak-PRFs, and the second LPN-based construction. We only briefly sketch the two DDH-based ones here without the tedious descriptions of HPS and key-homomorphic weak-PRFs.

Firstly, consider the probabilistic MAC constructions from labeled hash proof systems (HPS) in [Dodis et al. 2012]. We recall the notions about HPS and related constructions in Appendix B.

When instantiating it directly with the universal$_2$ HPS by Cramer and Shoup [Cramer and Shoup 2002] without the modification done in [Dodis et al. 2012], we obtain a regular but not strong MAC.

Let $\mathbb{G}$ be a group of prime-order p and let g_1, g_2 be two independent generators of $\mathbb{G}$. Define $\mathcal{M} = \mathbb{Z}_p$, then

- $\mathsf{Gen}(1^k)$: Pick $mk = (x_1, x_2, y_1, y_2)$ randomly in $\mathbb{Z}_p^4$.
- $\mathsf{MAC.Sign}_{mk}(m)$: Pick r randomly in $\mathbb{Z}_p$, let $C = (u, v) = (g_1^r, g_2^r)$ and $K = u^{x_1 m + y_1} v^{x_2 m + y_2}$, then output $\sigma = (C, K)$.
- $\mathsf{MAC.Ver}_{mk}(m, \sigma)$: Parse σ as $((u, v), K)$ and output accept if and only if $K = u^{x_1 m + y_1} v^{x_2 m + y_2}$.

Theorem 23. *The above MAC scheme is regular but not strong one-time secure and secretly-detectable but not publicly-detectable under the DDH assumption on G.*

Proof. The regular security is directly from [Dodis et al. 2012]. Since given a mac value $\sigma = (C, K) = ((u, v), K)$ of m, one can generate another valid mac value σ' of m by randomly chooses a $r' \in \mathbb{Z}_p$ then let $\sigma' = ((u^{r'}, v^{r'}), K^{r'})$. The validity is obvious, in fact, σ' is the mac value of m under the randomness rr'. Thus, the scheme is not strongly secure.

The validity of σ' cannot be verified publicly given (m, σ, σ'). In fact, since a valid (m, σ') pair has the same distribution as (m, σ), if there is an algorithm $\mathcal{A}$ which can publicly verify the validity of σ' given (m, σ, σ') , then it can distinguish whether or not (m, σ') has the same distribution as (m, σ). Thus, we can construct an efficient DDH and random tuple distinguisher $\mathcal{D}$: given (g_1, g_2, g_3, g_4), randomly choose $(x_1, x_2, y_1, y_2) \in \mathbb{Z}_p^4$ and $m \in \mathbb{Z}_p$, let $u = g_1^r, v = g_2^r, K = u^{x_1 m + y_1} v^{x_2 m + y_2}$, $u' = g_3^r, v' = g_4^r, K' = u'^{x_1 m + y_1} v'^{x_2 m + y_2}$, and $\sigma = ((u, v), K), \sigma' = ((u', v'), K')$, run $\mathcal{A}$ on (m, σ, σ'). If σ' is valid, then output 1 to indicate DDH tuple, else output 0. It is obvious that if $\mathcal{A}$ wins then $\mathcal{D}$ wins, too.

It is interesting to note that a HPS is naturally a KEM, but the malleability of the HPS cannot yield RCCA security for the KEM, for example, in σ', the encapsulate key is not K anymore. However, when the HPS is used as a MAC, σ' is still a valid mac value of m.

Secondly, consider the DDH-based construction achieving full security from key-homomorphic weak-PRFs. Let $\mathbb{G}$ be a group of prime-order p and let g be a generator of $\mathbb{G}$. Define $\mathcal{M} = \{0, 1\}^k$, then

- $\mathsf{Gen}(1^k)$: Pick $mk = (x, x'_1, x'_2, ..., x'_k)$ randomly in $\mathbb{Z}_p^{k+1}$.
- $\mathsf{MAC.Sign}_{mk}(m)$: Pick r randomly in $\mathbb{Z}_p$, let $u = g^r$ and $w = u^{x + \sum x'_i m_i}$, then output $\sigma = (u, w)$.
- $\mathsf{MAC.Ver}_{mk}(m, \sigma)$: Parse σ as (u, w) and output accept iff $w = u^{x + \sum x'_i m_i}$.

Theorem 24. *The above MAC scheme is regular but not strong one-time secure and secretly-detectable but not publicly-detectable under the DDH assumption on G.*

Proof. The regular security is directly from [Dodis et al. 2012]. Since given a mac value $\sigma = (u, w)$ of m, one can generate another valid mac value σ' of m by randomly chooses a $r' \in \mathbb{Z}_p$ then let $\sigma' = (u^{r'}, w^{r'})$. The validity is obvious, in fact, σ' is the mac value of m under the randomness rr'. Thus, the scheme is not strongly secure.

The validity of σ' cannot be verified publicly given (m, σ, σ'). In fact, since a valid (m, σ') pair has the same distribution as (m, σ), if there is an algorithm $\mathcal{A}$ which can publicly verify the validity of σ' given (m, σ, σ') , then it can distinguish whether or not (m, σ') has the same distribution as (m, σ). Thus, we can construct an efficient DDH and random tuple distinguisher: given $(g_1, g_2 = g_1^x, g_3 = g_1^r, g_4)$, randomly choose $(x'_1, x'_2, ..., x'_k) \in \mathbb{Z}_p^k$ and $m \in \{0,1\}^k$, then it should be noted that $\sigma = (g_1, g_2 g_1^{\sum x'_i m_i})$ is a valid mac value of m under the key $mk = (x, x'_1, x'_2, ..., x'_k)$. Let $\sigma' = (g_3, g_4 g_3^{\sum x'_i m_i})$, run $\mathcal{A}$ on (m, σ, σ'). If σ' is valid, then output 1 to indicate DDH tuple, else output 0.

6.2 Instantiations of sd-RCCA secure hybrid encryptions

According the two different paradigms provided in Section 4, we find appropriate KEM and DEM parts to make the ciphertexts fit for the message space of the corresponding MAC scheme.

Firstly, follow CCA KEM + sd (but not pd)-RCCA DEM paradigm in Section 4, we instantiate the refined Cramer-Shoup hybrid scheme in [Shoup 2000] with our first MAC scheme to obtain our first sd-RCCA secure scheme.

Let $\mathbb{G}$ be a group of prime-order p and let g_1, g_2 be two independent generators of $\mathbb{G}$, TCR be a target collision resistant hash functions, and KDF be a key derivation function with proper domain and range. Define $\mathcal{M} = \mathbb{Z}_p$, then

– $\mathsf{Gen}(1^k)$: Pick $x_1, x_2, y_1, y_2, z_1, z_2$ randomly in $\mathbb{Z}_p^6$, then let $pk = (c, d, h) = (g_1^{x_1} g_2^{x_2}, g_1^{y_1} g_2^{y_2}, g_1^{z_1} g_2^{z_2})$, $sk = (x_1, x_2, y_1, y_2, z_1, z_2)$, output (pk, sk).

– $\mathsf{HPKE.Enc}_{pk}(m)$: Pick r, r' randomly in $\mathbb{Z}_p$, let $u = g_1^r$, $v = g_2^r$ and $\alpha = TCR(u, v)$, then let $w = (c^\alpha d)^r$, $K = h^r$ and $(dk, mk) = KDF(K)$, where $mk = (x'_1, x'_2, y'_1, y'_2)$, then let $e = dk + m$, $u' = g_1^{r'}$, $v' = g_2^{r'}$, and $w' = u'^{x'_1 e + y'_1} v'^{x'_2 e + y'_2}$, output $C = (u, v, w, e, u', v', w')$.

– $\mathsf{HPKE.Dec}_{sk}(C)$: Parse C as (u, v, w, e, u', v', w'), let $\alpha = TCR(u, v)$, $K = h^r$ and $(dk, mk) = KDF(K)$, parse mk as (x'_1, x'_2, y'_1, y'_2), output $m' = e - dk$ if and only if $w = u^{x_1 \alpha + y_1} v^{x_2 \alpha + y_2}$ and $w' = u'^{x'_1 e + y'_1} v'^{x'_2 e + y'_2}$.

For the naturalness, we note that HPSs are natural components for PKEs. Here shows that a slightly careless use of HPSs might result in sd-RCCA.

Now, consider some variants of this scheme.

For the HPS in the MAC, with the knowledge of r', it is possible to generate w' publicly. Then if $mk = (x'_1, x'_2, y'_1, y'_2)$ is not derived from K, but added in the sk, thus $(c', d') = (g_1^{x'_1} g_2^{x'_2}, g_1^{y'_1} g_2^{y'_2})$ must be added in pk, then the resulting scheme might be more natural as a PKE and can avoid the use of a KDF. However, the scheme is not secure any more, since the knowledge of r' allows one to generate the MAC value for any messages publicly. Our scheme provides a natural way to solve this problem, and can reduce the size of the public-key.

Another way to solve this is to use a common randomness, that is letting $r' = r$, so that without the knowledge of r it is impossible to generate valid (u', v', w'). However, this will directly results in CCA security, and is not suitable for our purpose. The original CCA secure Cramer-Shoup can be seen as such an scheme, and which further integrates w and w' by letting $\alpha = TCR(u, v, e)$.

This also somewhat explains the difficulty to build an RCCA secure PKE more efficient than existing CCA secure ones. The similar thing also happens to our second scheme.

If efficiency is in consideration for naturalness, then the scheme is much more efficient than the RCCA but not sd-RCCA secure, secretly rerandomizable double strand Cramer-Shoup in [Prabhakaran and Rosulek 2007], which uses dozens of group elements and exponentiation operations.

Secondly, follow the CCCA KEM + AE (OT DEM + regular MAC) paradigm, by using the CCCA-secure KEM in [Hofheinz and Kiltz 2007] with an authenticated encryption, we will instantiate a hybrid scheme, where our second DDH-based MAC is used.

Let $\mathbb{G}$ be a group of prime-order p and let g be a generator of $\mathbb{G}$, TCR be a target collision resistant hash functions, and KDF be a key derivation function with proper domain and range. Define $\mathcal{M} = \{0, 1\}^k$, then

- $\mathsf{Gen}(1^k)$: Pick x, y, z randomly in $\mathbb{Z}_p^3$, let $pk = (c, d, h) = (g^x, g^y, g^z)$, $sk = (x, y, z)$, output (pk, sk).
- $\mathsf{HPKE.Enc}_{pk}(m)$: Pick r, r' randomly in $\mathbb{Z}_p$, let $u = g^r$, $w = (c^\alpha d)^r$ where $\alpha = TCR(u)$, then let $K = h^r$ and $(dk, mk) = KDF(K)$, where $mk = (x', x'_1, x'_2, ..., x'_k)$, then let $e = dk \oplus m$, $u' = g^{r'}$, and $w' = u'^{x + \sum x'_i e_i}$, output $C = (u, w, e, u', w')$.
- $\mathsf{HPKE.Dec}_{sk}(C)$: Parse C as (u, w, e, u', w') and let $\alpha = TCR(u)$, $K = h^r$ and $(dk, mk) = KDF(K)$, parse mk as $(x', x'_1, x'_2, ..., x'_k)$, output $m' = e \oplus dk$ if and only if $w = u^{x\alpha + y}$ and $w' = u'^{x' + \sum x'_i e_i}$.

Our sd-RCCA secure schemes are less efficient than existing CCA secure ones. In fact, in an efficient CCA secure hybrid encryption scheme, it is often the case that the KEM ciphertext is deterministically related to the encapsulation key, which makes it impossible to achieve RCCA security. However, regular MACs more efficient than strong ones still bring us a light.

7 Conclusion

We introduce regular (but not strong) probabilistic MACs into KEM+DEM style hybrid paradigm to construct sd-RCCA secure public-key encryptions. We show two examples of such MACs under the DDH assumption based on the work in [Dodis et al. 2012]. Instantiating proper DDH-based hybrid encryptions with these MACs, we obtain "natural" instances of sd-RCCA secure ones. This solves an open problem left in [Canetti and Krawczky 2003].

Acknowledgment

This work is supported by the National Natural Science Foundations of China (Nos. 61402353, 61373172) and the Fundamental Research Funds for the Central Universities (No.GK201603084).

References

[Abe et al. 2008] Abe, M., Gennaro, R., Kurosawa, K.: "Tag-KEM/DEM: A new framework for hybrid encryption"; Journal of Cryptology, 2008, 21, 1 (2008), 97-130.

[Alwen et al. 2014] Alwen, J., Hirt, M., Maurer, U., Patra, A., Raykov, P.: "Key-indistinguishable message authentication codes"; Security and Cryptography for Networks - SCN 2014, Springer International Publishing (2014), 476-493.

[An et al. 2002] An, J.H., Dodis, Y., Rabin, T.: "On the security of joint signature and encryption"; Advances in Cryptology-EUROCRYPT 2002, LNCS, vol. 2332, Springer, Heidelberg (2002), 83-107.

[Bellare et al. 2004] Bellare, M., Goldreich, O., Mityagin, A.: "The Power of Verification Queries in Message Authentication and Authenticated Encryption"; IACR Cryptology ePrint Archive, Report 2004/309(2004), https://eprint.iacr.org/2004/309

[Canetti and Krawczky 2003] Canetti, R., Krawczyk, H., Nielsen, J. B.: "Relaxing chosen-ciphertext security"; Advances in Cryptology-CRYPTO 2003, Springer, Berlin Heidelberg (2003), 565-582.

[Chen and Dong 2014] Chen,Y., Dong, Q.: "RCCA security for KEM+DEM style hybrid encryptions and a general hybrid paradigm from RCCA-secure KEMs to CCA-secure encryptions"; SECURITY AND COMMUNICATION NETWORKS, 2014, 7, 8 (2014), 1219-1231.

[Coretti et al. 2013] Coretti, S., Maurer, U., Tackmann, B.: "Constructing confidential channels from authenticated channelsPublic-key encryption revisited"; Advances in Cryptology-ASIACRYPT 2013, Springer, Berlin Heidelberg (2013), 134-153.

[Cramer and Shoup 2003] Cramer, R., Shoup, V.: "Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack"; SIAM Journal on Computing, 33, 1 (2003), 167-226.

[Cramer and Shoup 2002] Cramer, R., Shoup, V.: "Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption"; Advances in Cryptology-EUROCRYPT 2002, Springer, Berlin Heidelberg (2002), 45-64.

[Dodis et al. 2012] Dodis, Y., Kiltz, E., Pietrzak, K., Wichs, D.: "Message authentication, revisited"; Advances in Cryptology-EUROCRYPT 2012, Springer, Berlin Heidelberg (2012), 355-374.

[El Bansarkhani et al. 2014] El Bansarkhani, R., Dagdelen, Ö., Buchmann, J.: "Augmented learning with errors: The untapped potential of the error term"; Cryptology ePrint Archive, Report 2014/733 (2014), https://eprint.iacr.org/2014/733

[Gröth 2004] Gröth, J.: "Rerandomizable and replayable adaptive chosen ciphertext attack secure cryptosystems"; Proceedings of TCC 2004, LNCS, vol. 2951, Springer, Heidelberg (2004), 152-170.

[Hofheinz and Kiltz 2007] Hofheinz, D., Kiltz, E.: "Secure hybrid encryption from weakened key encapsulation"; Advances in Cryptology-CRYPTO 2007, Springer, Berlin Heidelberg (2007), 553-571.

[Hohenberger et al. 2012] Hohenberger, S., Lewko, A., Waters, B.: "Detecting dangerous queries: A new approach for chosen ciphertext security"; Advances in Cryptology-EUROCRYPT 2012, Springer, Berlin Heidelberg (2012), 663-681.

[Krawczyk 2001] Krawczyk, H.: "The order of encryption and authentication for protecting communications (or: How secure is SSL?)"; Advances in Cryptology-CRYPTO 2001, Springer, Berlin Heidelberg (2001), 310-331.

[Li et al. 2019] Li, Y., Yu, Y., Susilo, W., Min, G., Ni, J., Choo, R.: "Fuzzy Identity-Based Data Integrity Auditing for Reliable Cloud Storage Systems". IEEE Trans. on Dependable and Secure Computing, 16(1) (2019), 72-83.

[Maurer et al. 2012] Maurer, U., Rüedlinger, A., Tackmann, B.: "Confidentiality and integrity: A constructive perspective"; Theory of Cryptography-TCC 2012, Springer, Berlin Heidelberg (2012), 209-229.

[Prabhakaran and Rosulek 2007] Prabhakaran, M. M., Rosulek, M.: "Rerandomizable RCCA Encryption"; Proceedings of CRYPTO 2007, LNCS, vol. 4622, Springer, Heidelberg (2007), 517-534

[Shoup 2004] Shoup, V.: "ISO 18033-2: An emerging standard for public-key encryption"; Final Committee Draft (Dec 2004).

[Shoup 2000] Shoup, V.: "Using hash functions as a hedge against chosen ciphertext attack"; Advances in Cryptology-EUROCRYPT 2000, Springer, Berlin Heidelberg (2000), 275-288.

[Yu et al. 2017] Yu, Y., Au, M. H., Ateniese, G., Huang, X., Susilo, W., Dai, Y., Min, G.: "Identity-Based Remote Data Integrity Checking with Perfect Data Privacy Preserving for Cloud Storage". IEEE Trans. Information Forensics and Security, 12(4) (2017), 767-778.

A Sd-RCCA secure hybrid encryptions from CCCA secure KEMs and sd-RCCA secure AEs

Definition 25 (CCCA for KEMs). This is defined very similar to CCA security for KEMs, with the modification that in step 3, the decryption oracle is constrained, denoted by $\mathsf{KEM.Dec}_{sk}(\cdot,\cdot)$, which takes a predicate $pred : \mathcal{K} \to \{0,1\}$ and a ciphertext c as inputand returns a response as follows:

$$\mathsf{KEM.Dec}_{sk}(pred(\cdot), c) = \begin{cases} dk, \text{ If } \mathsf{KEM.Dec}_{sk}(c) = dk \neq \perp \;\; and \; pred(K) = 1 \\ \perp, \text{ Otherwise} \end{cases}$$

For an adversary $\mathcal{A}$ and an environment $\mathcal{E}$, a parameter called *plaintext uncertainty* $uncert_{A_E}(k)$ is defined by

$$uncert_{\mathcal{A},\mathcal{E}}(k) = \frac{1}{Q} \sum_{1 \le i \le Q} \Pr_{dk \in \mathcal{K}_K} [pred_i(dk) = 1 \; when \; \mathcal{A} \; runs \; with \; \mathcal{E}],$$

where $pred_i$ is the predicate $\mathcal{A}$ submits in its ith decapsulation query, and Q is the number of decapsulation queries $\mathcal{A}$ makes. A CCCA adversary $\mathcal{A}$ is valid if

1. $\mathcal{A}$ is PPT.

2. For all environment $\mathcal{E}$ running in less time than t_{CCCA}, $uncert_{\mathcal{A},\mathcal{E}}(k)$ is negligible in k, where t_{CCCA} is the runtime of the CCCA game excluding that of $\mathcal{A}$ and that for evaluating predicts.

Then a KEM $\mathcal{KEM}$ is CCCA secure if for every valid $\mathcal{A}$, its advantage $Adv^{\mathsf{ccca}}_{\Pi,\mathcal{A}}(k)$ (defined similar as for CCA security)is negligible in k.

For a CCCA attacker $\mathcal{A}_K$, we also denote the maximum of the $uncert_{\mathcal{A},\mathcal{E}}(k)$ over all environment with $t_{\mathcal{E}} \leq t_{CCCA}$ as $uncert_{\mathcal{A}_K}(k)$, which is negligible in k for all valid $\mathcal{A}_K$.

Proof of Theorem 4. The proof for not pd-RCCA security is almost the same as for Theorem 2, so we just prove the sd-RCCA security here.

Let $\equiv_{dk}$ be a compatible relation for $\mathcal{AE}$, we define a compatible relation for $\mathcal{HPKE}$ as follows: $(\psi, \chi) \equiv_{sk} (\psi', \chi')$ if $\psi = \psi'$ and $\chi \equiv_{dk} \chi'$ where $dk = \mathsf{KEM.Dec}_{sk}(\psi) = \mathsf{KEM.Dec}_{sk}(\psi')$. It is straightforward to verify $\equiv_{sk}$ is compatible for $\mathcal{HPKE}$ as long as $\equiv_{dk}$ is compatible for $\mathcal{AE}$.

Now, let $\mathcal{H}$ be an adversary playing $\mathsf{sd\text{-}RGAME.PKE}$. Let (ψ^*, χ^*) be the challenge ciphertext, dk^* is the encapsulated key in ψ^*. We modify the game by using a random key dk^+ in place of dk^* in both the encryption and decryption oracle, i.e, dk^+ is used to form the challenge ciphertext, and a decryption oracle query is replied by using dk^+ whenever dk^* should be used. Call this game $\mathsf{sd\text{-}RGAME.PKE}'$. Let F and F' be events that $\tilde{b} = b$ in $\mathsf{sd\text{-}RGAME.PKE}$ and $\mathsf{sd\text{-}RGAME.PKE}'$, respectively. Then

$$Adv^{\mathrm{sd-rcca}}_{\mathcal{HPKE},\mathcal{H}}(k) = |\Pr[F] - \frac{1}{2}|,$$

and we claim that

Lemma 26. *$|\Pr[F] - \Pr[F']| \leq 2Adv^{\mathrm{ccca}}_{\mathcal{KEM},\mathcal{A}_K}(k)$ for some valid CCCA adversary $\mathcal{A}_K$, which has $uncert_{\mathcal{A}_K}(k) = 2Adv^{\mathrm{rot}}_{\mathcal{AE},\mathcal{B}_A}(k)$ for some $\mathcal{B}_A$.*

The proof of this lemma is followed later.

We modify the game $\mathsf{sd\text{-}RGAME.PKE}'$ further by rejecting all ciphertext (ψ^*, χ) with $\chi \not\equiv_{dk^+} \chi^*$. Call this game $\mathsf{sd\text{-}RGAME.PKE}''$. Let F'' be events that $\tilde{b} = b$ in $\mathsf{sd\text{-}RGAME.PKE}''$, respectively. Since ψ^* is independent of dk^+, the authenticity of $\mathcal{AE}$ implies

$$|\Pr[F''] - \Pr[F']| \leq Q \cdot Adv^{\mathrm{rot}}_{\mathcal{AE},\mathcal{A}_A}(k)$$

where Q is the number of decryption queries made by $\mathcal{H}$, and $\mathcal{A}_A$ just uniformly choose one of the AE part of decryption queries made by $\mathcal{H}$ to submit to its decrypt-or-reject oracle.

Finally, we argue that $\mathcal{H}$ in $\mathsf{sd\text{-}RGAME.PKE}''$ in fact conducts an attack against the sd-ROT security of AE, i.e. $|\Pr[F''] - \frac{1}{2}| = Adv^{\text{sd-rot}}_{\mathcal{AE},\mathcal{A}'_A}(k)$, where $\mathcal{A}'_A$ is constructed as follows. $\mathcal{A}'_A$ first runs $\mathsf{PKE.Gen}$ to generate (pk, sk), then sends pk to $\mathcal{H}$. After $\mathcal{H}$ chooses its (m_0, m_1), $\mathcal{A}'_A$ gives them to its own encryption oracle and gets χ^*. Then $\mathcal{A}'_A$ runs $\mathsf{KEM.Enc}$ to generate (dk^*, ψ^*), and gives (ψ^*, χ^*) to $\mathcal{H}$. It should be noticed that now the key dk^+ used in encryption oracle of $\mathsf{GAME.AE}$ is chosen randomly from $\mathcal{K}_D$, so is independent of dk^*.

$\mathcal{A}'_A$ answers $\mathcal{H}$'s decryption query (ψ, χ) as follows:

– If $\psi = \psi^*$ and so that $\chi \neq \chi^*$, then $\mathcal{A}'_A$ forwards χ to its own decryption oracle, and returns the result to $\mathcal{H}$.

– If $\psi \neq \psi^*$, then $\mathcal{A}'_A$ uses sk to decrypt ψ.

When $\mathcal{H}$ outputs $\tilde{b}$, $\mathcal{A}'_A$ outputs $\tilde{b}$, too. $\mathcal{A}'_A$ perfectly simulates the game $\mathsf{sd\text{-}RGAME.PKE}''$, and $\mathcal{A}'_A$ wins if $\mathcal{H}$ does. So, $|\Pr[F''] - \frac{1}{2}| = Adv^{\text{sd-rot}}_{\mathcal{AE},\mathcal{A}'_A}(k)$.

Collecting all the probability proves the theorem.

Proof of Lemma 3 We show there is a CCCA adversary $\mathcal{A}_K$ against the underlying KEM scheme by using $\mathcal{H}$.

$\mathcal{A}_K$ asks to obtain the challenge (pk, dk_δ, ψ^*) in $\mathsf{GAME.KEM}$, then sends pk to $\mathcal{H}$. After $\mathcal{H}$ chooses its m_0 and m_1, $\mathcal{A}_K$ randomly chooses $b \in \{0, 1\}$, computes $\chi^* = \mathsf{DEM.Enc}_{dk_\delta}(m_b)$, and sends (ψ^*, χ^*) to $\mathcal{H}$.

To answer $\mathcal{H}$'s i-th decryption query (ψ_i, χ_i), $\mathcal{A}_K$ defines $pred_i : \mathcal{K}_K \to \{0, 1\}$ as follows:

$$pred_i(dk) = \begin{cases} 0, \text{If } \mathsf{AE.Dec}_{dk}(\chi_i) = \perp \ \ or \ \chi_i \equiv_{dk_\delta} \chi^* \\ 1, \text{Otherwise} \end{cases}$$

Clearly, $pred_i$ is efficiently computable:

– If $\psi_i = \psi^*$ then

  - If $\chi \equiv_{dk_\delta} \chi^*$ then $\mathcal{A}_K$ returns 'test' (note that $\mathcal{A}_K$ knows dk_δ).

  - Else $\mathcal{A}_K$ returns $\perp$.

– If $\psi_i \neq \psi^*$ then $\mathcal{A}_K$ queries $(pred_i, \chi_i)$ to its own oracle $\mathsf{KEM.Dec}_{sk}(\cdot, \cdot)$,

  - If $\perp$ is returned, then $\mathcal{A}_K$ returns $\perp$ to $\mathcal{H}$.

  - If dk is returned, then $\mathcal{A}_K$ uses this dk to decrypt χ, and returns the result to $\mathcal{H}$.

This perfectly simulates the decryption oracle for $\mathcal{H}$. When $\mathcal{H}$ outputs $\tilde{b}$, $\mathcal{A}_K$ checks whether or not $\tilde{b} = b$, if so it outputs $\tilde{\delta} = 1$, else outputs $\tilde{\delta} = 0$. And we have $\Pr[\tilde{b} = b|\delta = 1] = \Pr[F]|$, and $\Pr[\tilde{b} = b|\delta = 0] = \Pr[F']$, then

$$2Adv^{\text{ccca}}_{\mathcal{KEM},\mathcal{A}_K}(k) = |\Pr[F] - \Pr[F']|.$$

It is left to show that $uncert_{\mathcal{A}_K}(k) = 2Adv^{\mathrm{rot}}_{\mathcal{AE},\mathcal{B}_A}(k)$ for some $\mathcal{B}_A$.

We build $\mathcal{B}_A$ against the ROT security of $\mathcal{AE}$ as follows. $\mathcal{B}_A$ inputs 1^k and internally simulates an interaction between $\mathcal{A}_K$ and $\mathcal{H}$ as above completely faithful (the sk is known to $\mathcal{B}_A$). In this process, $\mathcal{B}_A$ randomly picks an index $j^* \in \{1, \dots, Q\}$, where Q is the number of decryption oracle queries made by $\mathcal{H}$. When A_K makes its j^*-th decryption query (ψ_{j^*}, χ_{j^*}), $\mathcal{B}_A$ submits χ_{j^*} to its own oracle, and outputs $\tilde{b} = 0$ iff $\perp$ is returned.

Note that our $\mathcal{B}_A$ never ask for its challenge ciphertext χ^*, just make its guess by an query to its own decrypt-or-reject oracle, so 'test' cases never happen.

Now, if $b = 0$ then $\mathcal{B}_A$ always output $\tilde{b} = 0$. And in case of $b = 1$, $\tilde{b} = 1$ iff χ_{j^*} is valid, that is, for an independently random key dk (used in the ROT game for $\mathcal{AE}$), $\mathsf{AE.Dec}_{dk}(\chi_{j^*}) \neq \perp$. So

$$\begin{aligned} Adv^{\mathrm{rot}}_{\mathcal{AE},\mathcal{B}_A}(k) &= |\frac{1}{2}\Pr[\tilde{b} = b|b = 0] - \frac{1}{2}\Pr[\tilde{b} = b|b = 1] - \frac{1}{2}| \\ &= \frac{1}{2}\Pr[\tilde{b} = b|b = 1] = \frac{1}{2}\Pr[dk \leftarrow \mathcal{K}_D : \mathsf{AE.Dec}_{dk}(\chi_{j^*}) \neq \perp] \\ &= \frac{1}{2Q_{\mathcal{A}}} \sum_{1 \le j^* \le Q} \Pr[pred_{j^*}(dk) = 1] \le \frac{1}{2} uncert_{\mathcal{A}_K}(k). \end{aligned}$$

B HPS related notions and the label HPS based MAC

Let us first recall labeled hash proof systems (HPS) in [Cramer and Shoup 2002].

Let $\mathcal{C}$, $\mathcal{K}$ be sets, $\mathcal{V} \subset \mathcal{C}$ a language. In the setting of PKEs, a (labeled) HPS can be viewed as a (labeled) KEM with some special properties. One can think of $\mathcal{C}$ as the sets of all possible ciphertexts, $\mathcal{V} \subset \mathcal{C}$ as the set of all valid ciphertexts, and $\mathcal{K}$ as the set of all possible encapsulated keys.

Let $\Lambda^l_k : \mathcal{C} \times \mathcal{L} \to \mathcal{K}$ be a labeled hash function indexed with $k \in \mathcal{SK}$ and label $l \in \mathcal{L}$, where $\mathcal{SK}$ and $\mathcal{L}$ are sets. Λ^l_k is *projective* if there exists a projection $\mu : \mathcal{SK} \to \mathcal{PK}$, such that $\mu(k) \in \mathcal{PK}$ defines the action of Λ^l_k over the subset $\mathcal{V}$. That is, for every $C \in \mathcal{V}$, the value $K = \Lambda^l_k(C)$ is uniquely determined by $\mu(k)$ and C. In contrast, nothing is guaranteed for $C \in \mathcal{C} \setminus \mathcal{V}$, and it might not be possible to compute $\Lambda^l_k(C)$ from $\mu(k)$ and C, but $\Lambda^l_k(C)$ can be computed from k and C (which is denoted by *extracting* in [Dodis et al. 2012]).

A projective hash function is *universal*$_2$ if for all C, $C^* \in \mathcal{C} \setminus \mathcal{V}$ with $C \neq C^*$, l,$l^* \in \mathcal{L}$ with $l \neq l^*$,

$$(\mu(k), \Lambda^l_k(C^*), \Lambda^l_k(C)) = (\mu(k), \Lambda^l_k(C^*), K)$$

for randomly chosen k and K.

A labelled HPS $\mathcal{HPS}$ consists of three algorithms($\mathsf{HPS.Param}$, $\mathsf{HPS.Pub}$, $\mathsf{HPS.Priv}$). Probabilistic $\mathsf{HPS.Param}$ that on input 1^k outputs instances of *params* $=$ (*group*, $\mathcal{C}$, $\mathcal{V}$, $\mathcal{K}$, $\mathcal{PK}$, $\mathcal{SK}$, $\mathcal{L}$, $\Lambda^{(\cdot)}_{(\cdot)} : \mathcal{C} \to \mathcal{K}$, $\mu : \mathcal{SK} \to \mathcal{PK}$), where *group* may contain some additional structural parameters. Deterministic $\mathsf{HPS.Pub}$ that on input the projection key $\mu(k)$, $C \in \mathcal{V}$, a witness r of the fact $C \in \mathcal{V}$, and a label $l \in \mathcal{L}$, outputs $\Lambda_k^l(C)$. Deterministic $\mathsf{HPS.Priv}$ that on input $k \in \mathcal{SK}$ outputs $\Lambda_k^l(C)$, without knowing a witness. We further assume that μ is efficiently computable and that there are efficient algorithms given for sampling $k \in \mathcal{SK}$, sampling $C \in \mathcal{V}$ uniformly with a witness r.

As computational problem we require that the *subset membership problem* is hard in $\mathcal{HPS}$, that is, random $C \in \mathcal{V}$ are computationally indistinguishable from random $C' \in \mathcal{C} \setminus \mathcal{V}$.

The probabilistic MAC based on labeled HPS in [Dodis et al. 2012] uses the message as a label:

- $\mathsf{Gen}(1^k)$: Sample $k \in \mathcal{SK}$ and output $mk = k$.
- $\mathsf{MAC.Sign}_{mk}(m)$: Sample $C \in \mathcal{V}$, compute $K = \Lambda_k^m(C)$, and output $\sigma = (C, K)$.
- $\mathsf{MAC.Ver}_{mk}(m, \sigma)$: Parse σ as (C, K) and output accept iff $K = \Lambda_k^m(C)$.

The scheme is proved to be regular secure (even when the forger is given oracle access to $\mathsf{MAC.Sign}_{mk}(\cdot)$ for many times). In Section 5.1, we instantiate it directly with the universal$_2$ HPS by Cramer and Shoup to achieve our desired property. However, this is different with the example given in [Dodis et al. 2012], which has a variant form and is called "explicit rejection variant" and is given as follows.

Let $\mathbb{G}$ be a group of prime-order p and let g be a generator of $\mathbb{G}$. Define $\mathcal{M} = \mathbb{Z}_p$. Let $H : \mathbb{G}^2 \times \mathcal{M} \to \mathbb{Z}_p$ be a (target) collision resistant hash function, then

- $\mathsf{Gen}(1^k)$: Pick $mk = (\omega, x_1, x_2)$ randomly in $\mathbb{Z}_p^3$.
- $\mathsf{MAC.Sign}_{mk}(m)$: Pick r randomly in $\mathbb{Z}_p$, let $C = (u, v) = (g^r, u^\omega)$ and $K = u^{x_1 l + x_2}$, where $l = H(u, v, m)$, then output $\sigma = (C, K)$.
- $\mathsf{MAC.Ver}_{mk}(m, \sigma)$: Parse σ as $((u, v), K)$ and output accept iff $v = u^\omega$ and $K = u^{x_1 l + x_2}$, where $l = H(u, v, m)$.

For this scheme, it is impossible to generate another valid σ' for m given a valid σ, so it is not proper for our purpose.