No-Directional and Backward-Leak Uni-Directional Updatable Encryption Are Equivalent

Chen, Huanhuan; Fu, Shihui; Liang, Kaitai

Citation (APA)
Chen, H., Fu, S., & Liang, K. (2022). No-Directional and Backward-Leak Uni-Directional Updatable Encryption Are Equivalent. In V. Atluri, R. Di Pietro, C. D. Jensen, & W. Meng (Eds.), *Computer Security – ESORICS 2022 - 27th European Symposium on Research in Computer Security, Proceedings* (Part 1 ed., pp. 387-407). (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Vol. 13554). Springer. https://doi.org/10.1007/978-3-031-17140-6_19

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

No-Directional and Backward-Leak Uni-Directional Updatable Encryption Are Equivalent

Huanhuan Chen, Shihui Fu$^{(\boxtimes)}$, and Kaitai Liang

Delft University of Technology, Delft, The Netherlands
{h.chen-2,shihui.fu,kaitai.liang}@tudelft.nl

Abstract. Updatable encryption (UE) enables the cloud server to update the previously sourced encrypted data to a new key with only an update token received from the client. Two interesting works have been proposed to clarify the relationships among various UE security notions. Jiang (ASIACRYPT 2020) proved the equivalence of every security notion in the bi-directional and uni-directional key update settings and further, the security notion in the no-directional key update setting is strictly stronger than the above two. In contrast, Nishimaki (PKC 2022) proposed a new definition of uni-directional key update that is called the backward-leak uni-directional key update, and showed the equivalence relation by Jiang does not hold in this setting.

We present a detailed comparison of every security notion in the four key update settings and prove that the security in the backward-leak uni-directional key update setting is actually equivalent to that in the no-directional key update setting. Our result reduces the hard problem of constructing no-directional key update UE schemes to the construction of those with backward-leak uni-directional key updates.

Keywords: Updatable encryption · Key update · Security notion

1 Introduction

When a client stores encrypted data on a cloud server, a good way of key management is to change keys periodically, so as to resist the risk of key leakage. This process is referred to as key rotation, in which the core of the update relies on how to update the previous encrypted data to be decryptable by a new key. A possible way is to download and decrypt the encrypted data with the old key, and then encrypt the data with the new key and upload the new encrypted data again. But the download and upload process would be extremely expensive if there exists a considerable amount of data.

Updatable encryption (UE) [3] provides a practical solution to the above dilemma. Its core idea is that the client offers the cloud server the ability to update ciphertexts by the update tokens, with a requirement that the update token should not leak any information about the data. There are two flavors

of UE depending on if ciphertexts are needed in the generation of the update token. One is called ciphertext-dependent UE [2,3,5,6], in which clients need to download partial components of the encrypted data, called ciphertext header, from the cloud to generate the update token, and the token can only update the corresponding ciphertext. The other, more practical than the previous one, is known as ciphertext-independent UE [4,10–12,14,16], in which the token only depends on the old and the new keys, and a single token can update all existing ciphertexts. In this work, we only focus on the latter.

Security Notions. The security of UE schemes should be maintained even under a temporary corruption of keys and update tokens. The original UE construction and its security model against passive adversaries were proposed by Boneh et al. [3], where the adversary in the security game should specify the epoch keys it wishes to know before sending queries to the challenger. The confidentiality notions were further strengthened by [4,11,12], which attempts to capture the practical abilities of the adaptive attacker. The adversary can corrupt epoch keys and updated tokens at any time during the game as long as it does not trigger the trivial win conditions, which will be checked after the adversary submits its guessing bit (more details can be found in Sect. 2). A summary of existing confidentiality notions and their major difference is presented in Fig. 1.

	Challenge Input	Challenge Output
IND-Enc-notion [12]	$(\bar{m}_0, \bar{m}_1)$	$(\mathsf{Enc}(\bar{m}_0), \mathsf{Enc}(\bar{m}_1))$
IND-Upd-notion [12]	$(\bar{c}_0, \bar{c}_1)$	$(\mathsf{Upd}(\bar{c}_0), \mathsf{Upd}(\bar{c}_1))$
IND-UE-notion [4]	$(\bar{m}_0, \bar{c}_1)$	$(\mathsf{Enc}(\bar{m}_0), \mathsf{Upd}(\bar{c}_1))$

Fig. 1. A summary of confidentiality notions, where $\mathsf{notion} \in \{\mathsf{CPA}, \mathsf{CCA}\}$. The adversary in each confidentiality game provides two challenge inputs based on the oracles it has access to and tries to distinguish the challenge outputs.

Boyd et al. [4] proved that IND-UE-notion is strictly stronger even than the combination of the prior two (in Fig. 1) defined in [12]. They further proposed an integrity notion called IND-CTXT. Jiang [10] defined another integrity notion called IND-PTXT. In the CTXT game, the adversary tries to provide a valid ciphertext that is different from the ciphertexts obtained during the game by the challenger; while in the PTXT game, the adversary needs to provide a valid ciphertext, whose underlying plaintext has not been queried during the game. Those two integrity notions are similar to the integrity notions of symmetric encryption schemes, but the adversary is provided with oracles specified in UE and trivial win conditions are also checked after the adversary submits its forgery.

Hereafter by security notions, we mean the set of all confidentiality and integrity notions in [4] and [10]: {detIND-UE-CPA,randIND-UE-CPA,detIND-UE-CCA, randIND-UE-CCA, IND-CTXT, IND-PTXT}, where

det/rand denotes the ciphertext updates are deterministic or randomized, respectively.

Key Update Directions. The update token is generated by two successive epoch keys via the token generation algorithm, i.e., $\Delta_{e,e+1} = \mathsf{TokenGen}(k_e, k_{e+1})$ (defined in Sect. 2); therefore the adversary may derive one of the two successive keys from the other if the update token is known. Jiang [10] investigated three key update directions: bi-directional key updates in which both the old key k_e and the new key k_{e+1} can be derived from the other, uni-directional key updates in which only the new key k_{e+1} can be derived from the old key k_e but k_e cannot be derived from k_{e+1}, and no-directional key updates in which no keys in the two successive epoch keys can be derived from the other. The direction of key update affects the computation of leakage information known to the adversary, which in turn affects the computation of trivial win conditions as well as security notions. However, the main result in [10] shows that the security notions in the bi-directional key update setting and in the uni-directional key update setting are equivalence, while the security notions in the no-directional key update setting are strictly stronger.

Nishimaki [14] recently introduced a new definition of uni-directional key update that is called the backward-leak uni-directional key update for distinction, where the update direction is the opposite of the original uni-directional key update in [10] (called the forward-leak uni-directional key update for distinction). That is, the old key k_e can be derived from the new key k_{e+1}, but k_{e+1} cannot be derived from k_e. Nishimaki [14] demonstrated a contrasting conclusion that the security notions in the backward-leak uni-directional key update setting are not equivalent to those in the bi-directional directional key update setting.

But the relations among UE schemes in the four kinds of keys update settings have not been fully investigated yet. Thus, a natural interesting open problem that should be clear before any valuable constructions is as follows:

> *What are the relations among UE schemes in the bi-directional, forward-leak uni-directional, backward-leak uni-directional, and no-directional key update settings?*

Our Contributions. At first glance, one may think that UE schemes with no-directional key updates should be strictly strong than UE with all the other three key update directions, just as proved in [10] that no-directional key updates setting leaks less information about keys, tokens and ciphertexts than the bi-directional and forward-leak uni-directional key updates. However, our main result provides a surprising result that, for each security notion, no-directional key update UE schemes, which were believed to be strictly stronger than the other directional key update schemes, are actually equivalent to those with backward-leak uni-directional key updates.

Our main technique is to analyze the relations of the trivial win conditions for each security notion in different key update settings. As in other semantic security definitions, the adversary in the security game for UE is provided access to different oracles to capture realistic attack models. However, it may

lead to a trivial win if the adversary queries some combinations of oracles. Therefore, the bookkeeping technique was developed in [11,12] that tracks the leakage information of tokens, keys, and ciphertexts known to the adversary during the game and checks if those leakages may lead the adversary to trivially win after the adversary submits its guessing bit. The direction of key update affects the computation of leakage information and thus, we analyze the relations among UE schemes by analyzing the relations of trivial win conditions in different key update directions, especially the backward-leak uni-directional key update which was not covered by [10].

Based on our result, when analyzing the security notions, we can treat UE schemes with no-directional key updates as those with backward-leak uni-directional key updates. Currently, there are only two no-directional key update UE schemes in the literature: one is built on Ciphertext Puncturable Encryption [16] and the other is built on one-way functions and indistinguishability obfuscation [14]. Our result can eliminate the need for constructing UE schemes with no-directional key dates while also keeping security, since it is sufficient to construct UE schemes with backward-leak uni-directional key updates, which is much easier than the former. More related works are given in Appendix A.

2   Updatable Encryption

We review the syntax of UE and the confidentiality and integrity definitions.

Definition 1 ([11]). *A* UE *scheme includes a tuple of PPT algorithms* {UE.KG, UE.Enc, UE.Dec, UE.TG, UE.Upd} *that operate in epochs, starting from 0.*

- UE.KG(1^λ)*: the key generation algorithm outputs an epoch key* k_e.
- UE.Enc(k_e, m)*: the encryption algorithm takes as input an epoch key* k_e *and a message* m *and outputs a ciphertext* c_e.
- UE.Dec(k_e, c_e)*: the decryption algorithm takes as input an epoch key* k_e *and a ciphertext* c_e *and outputs a message* m′.
- UE.TG(k_e, k_{e+1})*: the token generation algorithm takes as input two epoch keys* k_e *and* k_{e+1} *and outputs a token* Δ_{e+1}.
- UE.Upd(Δ_{e+1}, c_e)*: the update algorithm takes as input a token* Δ_{e+1} *and a ciphertext* c_e *and outputs a ciphertext* c_{e+1}.

Correctness for UE means that any valid ciphertext and its updates should be decrypted to the correct message under the appropriate epoch key. The definitions of confidentiality and integrity for UE are given in Definition 2 and Definition 3, respectively. In general, the adversary in each security game is provided with access to different oracles, which enables it to obtain information about epoch keys, update tokens and ciphertexts from the challenger. In the challenge phase of the confidentiality game, the adversary submits a challenge message m̄ and a challenge ciphertext c̄ according to the information it already has and receives a ciphertext from the challenger, and its goal is to guess the received ciphertext is an encryption of the message of m̄ or an update of c̄. Then

the adversary can continue to query the oracles and eventually provides a guessing bit. In the integrity game, the goal of the adversary is to forge a new valid ciphertext. In both security games, some combinations of oracles may lead to a trivial win of the game for the adversary, so the challenger will check if those trivial win conditions are triggered during the game by a bookkeeping technique developed in [12].

An overview of the oracles that the adversary has access to is shown in Fig. 4, how to compute the leakage set and its extension are described in Sect. 2.1, and the trivial win conditions in different security games are presented in Sect. 2.2.

Definition 2 (Confidentiality, [4]). *Let* $\mathsf{UE} = \{\mathsf{UE.KG}, \mathsf{UE.Enc}, \mathsf{UE.Dec}\}$ *be an updatable encryption scheme. For* $\mathsf{notion} \in \{\mathsf{detIND\text{-}UE\text{-}CPA}, \mathsf{randIND\text{-}UE\text{-}CPA}, \mathsf{detIND\text{-}UE\text{-}CCA}, \mathsf{randIND\text{-}UE\text{-}CCA}\}$, *the* notion *advantage of an adversary* $\mathcal{A}$ *is defined as:* $\mathsf{Adv}^{\mathsf{notion}}_{\mathsf{UE},\mathcal{A}}(1^\lambda) = \left| \Pr[\mathsf{Exp}^{\mathsf{notion\text{-}1}}_{\mathsf{UE},\mathcal{A}} = 1] - \Pr[\mathsf{Exp}^{\mathsf{notion\text{-}0}}_{\mathsf{UE},\mathcal{A}} = 1] \right|$ *where the experiment* $\mathsf{Exp}^{\mathsf{notion\text{-}b}}_{\mathsf{UE},\mathcal{A}}$ *is given in Fig. 2 and Fig. 4, and* det *and* rand *denote the ciphertext update procedure is deterministic and randomized, respectively. We say a UE scheme is* notion *secure if* $\mathsf{Adv}^{\mathsf{notion}}_{\mathsf{UE},\mathcal{A}}(1^\lambda) \leq \mathsf{negl}(\lambda)$

Definition 3 (Integrity, [4,10]). *Let* $\mathsf{UE} = \{\mathsf{UE.KG}, \mathsf{UE.Enc}, \mathsf{UE.Dec}\}$ *be an updatable encryption scheme. For* $\mathsf{notion} \in \{\mathsf{INT\text{-}CTXT}, \mathsf{INT\text{-}PTXT}\}$, *the* notion *advantage of an adversary* $\mathcal{A}$ *is defined as:* $\mathsf{Adv}^{\mathsf{notion}}_{\mathsf{UE},\mathcal{A}}(1^\lambda) = \left| \Pr[\mathsf{Exp}^{\mathsf{notion}}_{\mathsf{UE},\mathcal{A}} = 1] \right|$ *where the experiment* $\mathsf{Exp}^{\mathsf{notion}}_{\mathsf{UE},\mathcal{A}}$ *is given in Fig. 3 and Fig. 4. We say a UE scheme is* notion *secure if* $\mathsf{Adv}^{\mathsf{notion}}_{\mathsf{UE},\mathcal{A}}(1^\lambda) \leq \mathsf{negl}(\lambda)$

$\mathsf{Exp}^{\mathsf{xxIND\text{-}UE\text{-}atk\text{-}b}}_{\mathsf{UE},\mathcal{A}}$:

1 : **do Setup**; $\mathsf{phase} \leftarrow 0$

2 : $\mathsf{b}' \leftarrow \mathcal{A}^{\mathcal{O}}(1^\lambda)$

3 : **if** $((\mathcal{K}^* \cup \mathcal{C}^* \neq \emptyset)$ **or** $(\mathsf{xx} = \mathsf{det}$ and

4 : $(\tilde{\mathsf{e}} \in \mathcal{T}^*$ **or** $\mathcal{O}.\mathsf{Upd}(\bar{c})$ is required$)))$ **then**

5 : $\mathsf{twf} \leftarrow 1$

6 : **if** $\mathsf{twf} = 1$ **then**

7 : $\mathsf{b}' \xleftarrow{\$} \{0,1\}$

8 : **return** b'

Fig. 2. Generic description of the confidentiality experiment $\mathsf{Exp}^{\mathsf{xxIND\text{-}UE\text{-}atk\text{-}b}}_{\mathsf{UE},\mathcal{A}}$ for $\mathsf{xx} \in \{\mathsf{rand}, \mathsf{det}\}$, $\mathsf{atk} \in \{\mathsf{CPA}, \mathsf{CCA}\}$ and $\mathsf{b} \in \{0,1\}$. The flag $\mathsf{phase} \in \{0,1\}$ denotes whether or not $\mathcal{A}$ has queried the $\mathcal{O}.\mathsf{Chall}$ oracle, and twf tracks if the trivial win conditions are triggered, and $\mathcal{O} = \mathcal{O}.\{\mathsf{Enc}, \mathsf{Next}, \mathsf{Upd}, \mathsf{Corr}, \mathsf{Chall}, \widetilde{\mathsf{UpdC}}\}$ is the set of oracles $\mathcal{A}$ can access to, which are defined in Fig. 4. When $\mathsf{atk} = \mathsf{CCA}$, the decryption oracle $\mathcal{O}.\mathsf{Dec}$ is also added to $\mathcal{O}$. The computation of $\mathcal{K}^*, \mathcal{T}^*, \mathcal{C}^*$ are discussed in Sect. 2.1.

$$\underline{\mathsf{Exp}_{\mathsf{UE},\mathcal{A}}^{\mathsf{IND\text{-}atk}}:}$$

1 :   **do Setup**; $\mathsf{win} \leftarrow 0$

2 : $\mathcal{A}^{\mathcal{O}}(1^{\lambda})$

3 :   **if** $\mathsf{twf} = 1$ **then**

4 : $\mathsf{win} \leftarrow 0$

5 :   **return** win

Fig. 3. Generic description of the confidentiality experiment $\mathsf{Exp}_{\mathsf{UE},\mathcal{A}}^{\mathsf{IND\text{-}atk}}$ for $\mathsf{atk} \in \{\mathsf{CTXT}, \mathsf{PTXT}\}$. The flag win tracks whether or not $\mathcal{A}$ provided a valid forgery, twf tracks if the trivial win conditions are triggered, and $\mathcal{O} = \mathcal{O}.\{\mathsf{Enc}, \mathsf{Next}, \mathsf{Upd}, \mathsf{Corr}, \mathsf{Try}\}$ is the set of oracles $\mathcal{A}$ can access to, which are defined in Fig. 4.

2.1   Leakage Sets

In security games of UE, the adversary is provided access to various oracles as shown in Fig. 4, so it can learn some information about update keys, epoch tokens and ciphertexts during the query phase. Moreover, it can extend the information via its known tokens, and the extension depends on the direction of key update and the direction of ciphertext update. We start by describing the leakage sets in [11,12], and then show how to compute the extended leakage sets in different key update direction settings.

Epoch Leakage Sets. We record the following epoch sets related to epoch keys, update tokens, and challenge-equal ciphertexts.

- $\mathcal{K}$: Set of epochs in which the adversary corrupted the epoch key from $\mathcal{O}.\mathsf{Corr}$.
- $\mathcal{T}$: Set of epochs in which the adversary corrupted the update token from $\mathcal{O}.\mathsf{Corr}$.
- $\mathcal{C}$: Set of epochs in which the adversary learned a challenge-equal ciphertext (the ciphertext related to the challenge inputs) from $\mathcal{O}.\mathsf{Chall}$ or $\mathcal{O}.\mathsf{Upd\widetilde{C}}$.

The adversary can use its corrupted tokens to extend $\mathcal{K}, \mathcal{T}, \mathcal{C}$ to infer more information. We use $\mathcal{K}^{*}, \hat{\mathcal{K}}^{*}, \mathcal{T}^{*}, \mathcal{C}^{*}$ as the extended sets respectively, and how to compute $\mathcal{K}^{*}, \hat{\mathcal{K}}^{*}, \mathcal{T}^{*}, \mathcal{C}^{*}$ are shown later.

Information Leakage Sets. We record the following sets related to ciphertexts known to the adversary.

- $\mathcal{L}$: Set of non-challenge equal ciphertexts $(c, \mathsf{c}, \mathsf{e}; \mathsf{m})$ the adversary learned from $\mathcal{O}.\mathsf{Enc}$ or $\mathcal{O}.\mathsf{Upd}$.
- $\widetilde{\mathcal{L}}$: Set of challenge-equal ciphertexts $(\tilde{\mathsf{c}}_{\mathsf{e}}, \mathsf{e})$ the adversary learned from $\mathcal{O}.\mathsf{Chall}$ or $\mathcal{O}.\mathsf{Upd\widetilde{C}}$.

The adversary can also use its corrupted tokens to extend $\mathcal{L}, \widetilde{\mathcal{L}}$ to infer more information about ciphertexts. In the deterministic update setting, we denote $\mathcal{L}^{*}, \widetilde{\mathcal{L}}^{*}$ as the extended sets of $\mathcal{L}$ and $\widetilde{\mathcal{L}}$, respectively.

Setup(1^λ):

$k_0 \xleftarrow{\$} \mathsf{UE.KG}(1^\lambda)$

$\Delta_0 \leftarrow \bot; \mathsf{e}, \mathsf{c}, \mathsf{twf} \leftarrow 0$

$\mathcal{L}, \tilde{\mathcal{L}}, \mathcal{C}, \mathcal{K}, \mathcal{T} \leftarrow \emptyset$

$\mathcal{O}.\mathsf{Enc}(\mathsf{m})$:

$c \leftarrow c + 1$

$\mathsf{c} \xleftarrow{\$} \mathsf{UE.ENC}(\mathsf{k_e}, \mathsf{m}))$

$\mathcal{L} \leftarrow \mathcal{L} \cup \{(c, \mathsf{c}, \mathsf{e}; \mathsf{m})\}$

return c

$\mathcal{O}.\mathsf{Dec}(\mathsf{c})$:

m' or $\bot \leftarrow \mathsf{UE.Dec}(\mathsf{k_e}, \mathsf{c})$

if $\Big((\mathsf{xx} = \mathsf{det}$ **and** $(\mathsf{c}, \mathsf{e}) \in \tilde{\mathcal{L}}^*)$ **or**

$\quad (\mathsf{xx} = \mathsf{rand}$ **and** $(\mathsf{m}', \mathsf{e}) \in \tilde{\mathcal{Q}}^*)\Big)$ **then**

$\quad \mathsf{twf} \leftarrow 1$

return m' or $\bot$

$\mathcal{O}.\mathsf{Next}()$:

$\mathsf{e} \leftarrow \mathsf{e} + 1$

$\mathsf{k_e} \xleftarrow{\$} \mathsf{UE.KG}(1^\lambda)$

$\Delta_\mathsf{e} \leftarrow \mathsf{UE.TG}(\mathsf{k_{e-1}}, \mathsf{k_e})$

if phase $= 1$ **then**

$\quad \tilde{\mathsf{c}}_\mathsf{e} \leftarrow \mathsf{UE.Upd}(\Delta_\mathsf{e}, \tilde{\mathsf{c}}_{\mathsf{e}-1})$

$\mathcal{O}.\mathsf{Corr}(\mathsf{inp}, \hat{\mathsf{e}})$

if $\hat{\mathsf{e}} > \mathsf{e}$ **then**

$\quad$ **return** $\bot$

if $\mathsf{inp} = \mathsf{key}$ **then**

$\quad \mathcal{K} \leftarrow \mathcal{K} \cup \{\hat{\mathsf{e}}\}$

$\quad$ **return** $\mathsf{k}_{\hat{\mathsf{e}}}$

if $\mathsf{inp} = \mathsf{token}$ **then**

$\quad \mathcal{T} \leftarrow \mathcal{T} \cup \{\hat{\mathsf{e}}\}$

$\quad$ **return** $\Delta_{\hat{\mathsf{e}}}$

$\mathcal{O}.\mathsf{Upd}(\mathsf{c}_{\mathsf{e}-1})$:

if $(j, \mathsf{c}_{\mathsf{e}-1}, \mathsf{e} - 1; \mathsf{m}) \notin \mathcal{L}$ **then**

$\quad$ **return** $\bot$

$\mathsf{c_e} \leftarrow \mathsf{UE.Upd}(\Delta_\mathsf{e}, \mathsf{c}_{\mathsf{e}-1})$

$\mathcal{L} \leftarrow \mathcal{L} \cup \{(j, \mathsf{c_e}, \mathsf{e}; \mathsf{m})\}$

$\mathcal{O}.\mathsf{Chall}(\bar{\mathsf{m}}, \bar{\mathsf{c}})$

if phase $= 1$ **then**

$\quad$ **return** $\bot$

phase $\leftarrow 1; \tilde{\mathsf{e}} \leftarrow \mathsf{e}$

if $(\cdot, \bar{\mathsf{c}}, \mathsf{e} - 1; \bar{\mathsf{m}}_1) \notin \mathcal{L}$ **then**

$\quad$ **return** $\bot$

if $b = 0$ **then**

$\quad \tilde{\mathsf{c}}_{\tilde{\mathsf{e}}} \leftarrow \mathsf{UE.Enc}(\mathsf{k}_{\tilde{\mathsf{e}}}, \bar{\mathsf{m}})$

else

$\quad \tilde{\mathsf{c}}_{\tilde{\mathsf{e}}} \leftarrow \mathsf{UE.Upd}(\Delta_{\tilde{\mathsf{e}}}, \bar{\mathsf{c}})$

$\mathcal{C} \leftarrow \mathcal{C} \cup \{\tilde{\mathsf{e}}\}$

$\bar{\mathcal{L}} \leftarrow \bar{\mathcal{L}} \cup \{(\tilde{\mathsf{c}}_{\tilde{\mathsf{e}}}, \tilde{\mathsf{e}})\}$

return $\tilde{\mathsf{c}}_{\tilde{\mathsf{e}}}$

$\mathcal{O}.\mathsf{Upd}\widetilde{\mathsf{C}}$

if phase $\neq 1$ **then**

$\quad$ **return** $\bot$

$\mathcal{C} \leftarrow \mathcal{C} \cup \{\mathsf{e}\}$

$\bar{\mathcal{L}} \leftarrow \bar{\mathcal{L}} \cup \{(\tilde{\mathsf{c}}_\mathsf{e}, \mathsf{e})\}$

return $\tilde{\mathsf{c}}_\mathsf{e}$

$\mathcal{O}.\mathsf{Try}(\widetilde{\mathsf{c}})$

m' or $\bot \leftarrow \mathsf{UE.Dec}(\mathsf{k_e}, \tilde{\mathsf{c}})$

if $(\mathsf{e} \in \mathcal{K}^*$ **or** $(\mathsf{atk} = \mathsf{CTXT}$ **and** $(\tilde{\mathsf{c}}, \mathsf{e}) \in \mathcal{L}^*)$ **or**

$\quad (\mathsf{atk} = \mathsf{PTXT}$ **and** $(\mathsf{m}', \mathsf{e}) \in \tilde{\mathcal{Q}}^*)\Big)$ **then**

$\quad \mathsf{twf} \leftarrow 1$

if $\mathsf{m}' \neq \bot$ **then**

$\quad \mathsf{win} \leftarrow 1$

Fig. 4. Oracles in the UE security games. m_1 is the underlying message of the challenge input ciphertext $\bar{\mathsf{c}}$. The leakage sets $\mathcal{L}, \widetilde{\mathcal{L}}, \mathcal{L}^*, \widetilde{\mathcal{L}}^*, \mathcal{C}, \mathcal{K}, \mathcal{K}^*, \mathcal{T}, \mathcal{T}^*, \mathcal{Q}, \mathcal{Q}^*, \widetilde{\mathcal{Q}}^*$ are defined in Sect. 2.1.

In randomized UE schemes, we use $\mathcal{Q}^*, \widetilde{\mathcal{Q}}^*$ to denote respectively the extended sets of $\mathcal{L}, \widetilde{\mathcal{L}}$:

– $\mathcal{Q}^*$: Set of plaintexts (m, e). The adversary learned in the query phase or could create a ciphertext of m in the epoch e.

– $\widetilde{\mathcal{Q}}^*$: Set of challenge plaintexts $\{(\bar{\mathsf{m}}, \mathsf{e}), (\bar{\mathsf{m}}_1, \mathsf{e})\}$, where $(\bar{\mathsf{m}}, \bar{\mathsf{c}})$ is the query input of $\mathcal{O}.\mathsf{Upd}$ and $\bar{\mathsf{m}}_1$ is the plaintext of $\bar{\mathsf{c}}$. The adversary learned in the query phase or could create a ciphertext of $\bar{\mathsf{m}}$ or $\bar{\mathsf{m}}_1$ in the epoch e.

Inferred Leakage Sets. The adversary can infer more information from $\mathcal{K}$, $\mathcal{L}$ and $\mathcal{C}$ via its corrupted tokens, which are computed as follows.

Key Leakage. Since the update tokens are generated from two successive epoch keys by $\Delta_{\mathsf{e}+1} = \mathsf{UE.TG}(\mathsf{k}_\mathsf{e}, \mathsf{k}_{\mathsf{e}+1})$, one epoch key in $\{\mathsf{k}_\mathsf{e}, \mathsf{k}_{\mathsf{e}+1}\}$ may be inferred by the other via the known token $\Delta_{\mathsf{e}+1}$.

– No-directional key updates: $\mathcal{K}^*_{\mathsf{no}} = \mathcal{K}$. The adversary does have more information about keys except $\mathcal{K}$, since tokens cannot be used to derive keys.
– Forward-leak uni-directional key updates:

$$\mathcal{K}^*_{\mathsf{f\text{-}uni}} = \{\mathsf{e} \in \{0, \ldots, l\} \mid \mathsf{CorrK}(\mathsf{e}) = \mathsf{true}\},$$

where $\mathsf{true} \leftarrow \mathsf{CorrK}(\mathsf{e}) \iff (\mathsf{e} \in \mathcal{K}) \vee (\mathsf{CorrK}(\mathsf{e}-1) \wedge \mathsf{e} \in \mathcal{T})$. The adversary can infer more keys from corrupted tokens and keys in the previous epoch.
– Backward-leak uni-directional key updates:

$$\mathcal{K}^*_{\mathsf{b\text{-}uni}} = \{\mathsf{e} \in \{0, \ldots, l\} \mid \mathsf{CorrK}(\mathsf{e}) = \mathsf{true}\}, \tag{1}$$

where $\mathsf{true} \leftarrow \mathsf{CorrK}(\mathsf{e}) \iff (\mathsf{e} \in \mathcal{K}) \vee (\mathsf{CorrK}(\mathsf{e}+1) \wedge \mathsf{e}+1 \in \mathcal{T})$. Keys can be inferred from corrupted tokens and keys in the next epoch.
– Bi-directional key updates:

$$\mathcal{K}^*_{\mathsf{bi}} = \{\mathsf{e} \in \{0, \ldots, l\} \mid \mathsf{CorrK}(\mathsf{e}) = \mathsf{true}\}, \tag{2}$$

where $\mathsf{true} \leftarrow \mathsf{CorrK}(\mathsf{e}) \iff (\mathsf{e} \in \mathcal{K}) \vee (\mathsf{CorrK}(\mathsf{e}-1) \wedge \mathsf{e} \in \mathcal{T}) \vee (\mathsf{CorrK}(\mathsf{e}+1) \wedge \mathsf{e}+1 \in \mathcal{T})$. Besides the corrupted keys $\mathcal{K}$, the adversary can infer more keys both from key upgrades and downgrades, i.e., $\mathcal{K}^*_{\mathsf{bi}} = \mathcal{K}^*_{\mathsf{f\text{-}uni}} \cup \mathcal{K}^*_{\mathsf{b\text{-}uni}}$.

In the integrity game, a set $\hat{\mathcal{K}}$ is defined to check if the adversary can trivially forge a valid ciphertext as follows.

$$\hat{\mathcal{K}}^* = \{\mathsf{i} \in \{0, \ldots, l\} \mid \mathsf{ForgK}(i) = \mathsf{true}\}$$
$$\mathsf{true} \leftarrow \mathsf{ForgK}(i) \iff (\mathsf{i} \in \mathcal{K}) \vee (\mathsf{CorrK}(\mathsf{e}-1) \wedge \mathsf{e} \in \mathcal{T}) \tag{3}$$

Token Leakage. The adversary knows a token by either corrupting or inferring from two successive keys. Then we have

$$\mathcal{T}^*_{\mathsf{kk}} = \{\mathsf{e} \in \{0, \ldots, l\} \mid (\mathsf{e} \in \mathcal{T}) \vee (\mathsf{e} \in \mathcal{K}^*_{\mathsf{kk}} \wedge \mathsf{e}-1 \in \mathcal{K}^*_{\mathsf{kk}}\}, \tag{4}$$

for $\mathsf{kk} \in \{\mathsf{no}, \mathsf{f\text{-}uni}, \mathsf{b\text{-}uni}, \mathsf{bi}\}$.

Ciphertext Leakages. Different from the direction of key update, ciphertexts should always be upgraded but are not necessarily downgraded by tokens, so there are only two types of ciphertext directions.

– Uni-directional ciphertext updates:

$$\mathcal{C}^*_{\mathsf{kk,uni}} = \{e \in \{0, \ldots, l\} \mid \mathsf{ChallEq}(e) = \mathsf{true}\}, \tag{5}$$

where $\mathsf{ChallEq}(e) = \mathsf{true} \iff (e \in \mathcal{C}) \vee (\mathsf{ChallEq}(e-1) \wedge e \in \mathcal{T}^*_{\mathsf{kk}})$. Besides the learned ciphertext $\mathcal{C}$, the adversary can infer more ciphertexts from corrupted tokens and ciphertexts in the previous epoch.

– Bi-directional ciphertext updates:

$$\mathcal{C}^*_{\mathsf{kk,bi}} = \{e \in \{0, \ldots, l\} \mid \mathsf{ChallEq}(e) = \mathsf{true}\}, \tag{6}$$

where $\mathsf{ChallEq}(e) = \mathsf{true} \iff (e \in \mathcal{C}) \vee (\mathsf{ChallEq}(e-1) \wedge e \in \mathcal{T}^*_{\mathsf{kk}}) \vee (\mathsf{ChallEq}(e+1) \wedge e+1 \in \mathcal{T}^*_{\mathsf{kk}})$. Besides the learned ciphertext $\mathcal{C}$, the adversary can infer more ciphertexts both from key upgrades and downgrades.

Remark 1. From the definition, the leakage sets have the following relations,

– $(\tilde{c}_e, e) \in \widetilde{\mathcal{L}} \iff e \in \mathcal{C}$,
– $(\tilde{c}_e, e) \in \widetilde{\mathcal{L}}^* \iff e \in \mathcal{C}^* \iff \{(\bar{m}, e), (\bar{m}_1, e)\} \in \widetilde{\mathcal{Q}}^*$.

An example of the computation of leakage sets is provided in Appendix B.

2.2 Trivial Win Conditions

In the security games of UE, the leaked information probably leads the adversary to trivially win the game. The challenger will check if any trivial win condition is triggered at the end of the game. A summary of trivial win conditions is described in Fig. 5, which follows from the analysis in [4,10–12]. A detailed explanation of the trivial win condition is provided in Appendix C.

notion	$\mathcal{K}^* \cap \mathcal{C}^* \neq \emptyset$	$\tilde{e} \in \mathcal{T}^*$ or $\mathcal{O}.\mathsf{Upd}(\bar{c})$ is queried	$(\bar{c}, e) \in \widetilde{\mathcal{L}}^*$	$(\bar{m}', e) \in \widetilde{\mathcal{Q}}^*$	$e \in \hat{\mathcal{K}}^*$	$(\bar{c}, e) \in \mathcal{L}^*$	$(\bar{m}', e) \in \mathcal{Q}^*$
detIND-UE-CPA	✓	✓	×	×	×	×	×
randIND-UE-CPA	✓	×	×	×	×	×	×
detIND-UE-CCA	✓	✓	✓	×	×	×	×
randIND-UE-CCA	✓	×	×	✓	×	×	×
IND-CTXT	×	×	×	×	✓	✓	×
IND-PTXT	×	×	×	×	✓	×	✓

Fig. 5. Trivial win conditions in different security games for updatable encryption. ✓ and × indicate whether the security notion considers the corresponding trivial win conditions or not. $\tilde{e}$ is the challenge epoch, i.e., the epoch the adversary queries $\mathcal{O}.\mathsf{Chall}$, and e represents the current epoch.

We review some properties of leakage sets as follows.

Definition 4 (Firewall, [4,10–12]). *An insulated region with firewalls* fwl *and* fwr, *denoted by* $\mathcal{FW}$, *is the consecutive sequence of epochs* $(\mathsf{fwl}, \ldots, \mathsf{fwr})$ *for which:*

– *no key in the sequence of epochs* $(\mathsf{fwl}, \ldots, \mathsf{fwr})$ *is corrupted;*
– *the tokens* Δ_{fwl} *and* $\Delta_{\mathsf{fwr}+1}$ *are not corrupted;*
– *all tokens* $\{\Delta_{\mathsf{fwl}+1}, \ldots, \Delta_{\mathsf{fwr}}\}$ *are corrupted.*

Denote the union of all firewalls as $\mathcal{IR} := \bigcup_{(\mathsf{fwl},\mathsf{fwr}) \in \mathcal{FW}} \{\mathsf{fwl}, \ldots, \mathsf{fwr}\}$. The following lemma shows that $\mathcal{IR}$ is the complementary set of $\mathcal{K}_{\mathsf{bi}}^*$.

Lemma 1 (Lemma 3.1, [10]). *For any* $\mathcal{K}, \mathcal{T} \in \{0, \ldots, l\}$, *we have* $\mathcal{K}_{\mathsf{bi}}^* = \{0, \ldots, l\} \setminus \mathcal{IR}$, *where* l *is the maximal number of updates.*

Corollary 1. *Since* $\mathcal{K}_{\mathsf{bi}}^* = \mathcal{K}_{\mathsf{f\text{-}uni}}^* \cup \mathcal{K}_{\mathsf{b\text{-}uni}}^*$ *by definition, we have* $\{0, \ldots, l\} = \mathcal{IR} \cup \mathcal{K}_{\mathsf{f\text{-}uni}}^* \cup \mathcal{K}_{\mathsf{b\text{-}uni}}^*$.

Remark 2. For an epoch $\mathsf{e} \in \mathcal{K}_{\mathsf{f\text{-}uni}}^*$, it holds that either $\mathsf{e} \in \mathcal{K}$ or there exists an epoch $\mathsf{e_f}$ before e, such that $\mathsf{e_f} \in \mathcal{K}$ and $\{\mathsf{e_f}, \ldots, \mathsf{e}\} \in \mathcal{T}$; for an epoch $\mathsf{e} \in \mathcal{K}_{\mathsf{b\text{-}uni}}^*$, it holds that either $\mathsf{e} \in \mathcal{K}$ or there exists an epoch $\mathsf{e_b}$ after e, such that $\mathsf{e_b} \in \mathcal{K}$ and $\{\mathsf{e}, \ldots, \mathsf{e_b}\} \in \mathcal{T}$. That follows directly from the definition.

3   Relations Among Security Notions

To capture the security for UE schemes with kk-directional key updates and cc-directional ciphertext updates, we consider the $(\mathsf{kk}, \mathsf{cc})$-variant of each security notion as defined in [10] (where $\mathsf{kk} \in \{\mathsf{bi}, \mathsf{f\text{-}uni}, \mathsf{b\text{-}uni}, \mathsf{no}\}$ and $\mathsf{cc} \in \{\mathsf{uni}, \mathsf{bi}\}$), and then compare the relations among all the variants of each security notion.

Definition 5 ($(\mathsf{kk}, \mathsf{cc})$-variant of confidentiality, [10]). *Let* $\mathsf{UE} = \{\mathsf{UE.KG},$ $\mathsf{UE.Enc}, \mathsf{UE.Dec}\}$ *be an updatable encryption scheme. For* notion $\in$ $\{\mathsf{detIND\text{-}UE\text{-}CPA}, \mathsf{randIND\text{-}UE\text{-}CPA}, \mathsf{detIND\text{-}UE\text{-}CCA}, \mathsf{randIND\text{-}UE\text{-}CCA}\}$, *the* $(\mathsf{kk}, \mathsf{cc})$-notion *advantage of an adversary* $\mathcal{A}$ *is defined as*

$$\mathsf{Adv}_{\mathsf{UE}, \mathcal{A}}^{(\mathsf{kk},\mathsf{cc})\text{-notion}}(1^\lambda) = \left| \Pr[\mathsf{Exp}_{\mathsf{UE}, \mathcal{A}}^{(\mathsf{kk},\mathsf{cc})\text{-notion-1}} = 1] - \Pr[\mathsf{Exp}_{\mathsf{UE}, \mathcal{A}}^{(\mathsf{kk},\mathsf{cc})\text{-notion-0}} = 1] \right|,$$

where the experiment $\mathsf{Exp}_{\mathsf{UE}, \mathcal{A}}^{(\mathsf{kk},\mathsf{cc})\text{-notion-b}}$ *is the same as the experiment* $\mathsf{Exp}_{\mathsf{UE}, \mathcal{A}}^{\mathsf{notion-b}}$ *(see Fig. 2 and Fig. 4), except all leakage sets are computed in the* kk-directional *key update setting and* cc-directional ciphertext update setting (see Sect. 2.1).

Definition 6 ($(\mathsf{kk}, \mathsf{cc})$-variant of integrity, [10]). *Let* $\mathsf{UE} = \{\mathsf{UE.KG},$ $\mathsf{UE.Enc}, \mathsf{UE.Dec}\}$ *be an updatable encryption scheme. For* notion $\in$ $\{\mathsf{INT\text{-}CTXT}, \mathsf{INT\text{-}PTXT}\}$, *the* $(\mathsf{kk}, \mathsf{cc})$-notion *advantage of an adversary* $\mathcal{A}$ *is defined as*

$$\mathsf{Adv}_{\mathsf{UE}, \mathcal{A}}^{(\mathsf{kk},\mathsf{cc})\text{-notion}}(1^\lambda) = \left| \Pr[\mathsf{Exp}_{\mathsf{UE}, \mathcal{A}}^{(\mathsf{kk},\mathsf{cc})\text{-notion}} = 1] \right|,$$

where the experiment $\mathsf{Exp}_{\mathsf{UE},\mathcal{A}}^{(\mathsf{kk},\mathsf{cc})\text{-notion}}$ *is the same as the experiment* $\mathsf{Exp}_{\mathsf{UE},\mathcal{A}}^{\mathsf{notion}}$ *(see Fig. 3 and Fig. 4), except all leakage sets are computed in the* kk-*directional key update setting and* cc-*directional ciphertext update setting (see Sect. 2.1).*

A general idea to analyze the relation of any two out of the eight variants of each security notion is to construct a reduction $\mathcal{B}$, which runs the security experiment of one variant while simulating all the responses to the queries made by the adversary $\mathcal{A}$ in the security experiment of the other variant and forwards the guess result from $\mathcal{A}$ to its challenger. Recall that if a trivial win condition is triggered, the adversary will lose the game. Therefore, if the reduction $\mathcal{B}$ does not trigger trivial win conditions (when $\mathcal{A}$ does not), the advantage of the reduction $\mathcal{B}$ will be larger than that of the adversary $\mathcal{A}$. Thus, the relation of the two variants depends on the relation of trivial win conditions in each update direction setting.

3.1 Relations Among Confidentiality Notions

The relations of eight variants of confidentiality are shown in Fig. 6. We mainly prove the equivalence of each confidentiality notion in the bi-directional key update setting and backward-leak uni-directional key update setting, i.e., $(\mathsf{no},\mathsf{bi})$-notion $\Longleftrightarrow$ $(\mathsf{b\text{-}uni},\mathsf{bi})$-notion and $(\mathsf{no},\mathsf{uni})$-notion $\Longleftrightarrow$ $(\mathsf{b\text{-}uni},\mathsf{uni})$-notion. Then the rest of the relations in Fig. 6 can easily follow from the prior work in [10].
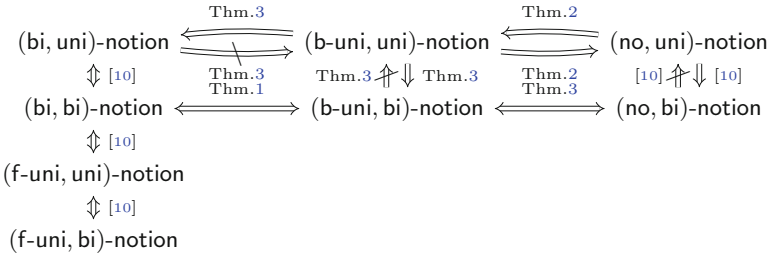

Fig. 6. Relations among the eight variants on confidentiality for notion $\in$ {detIND$-$UE-CPA, randIND-UE-CPA, detIND-UE-CCA, randIND-UE-CCA}.

The following two lemmas show UE schemes with bi-directional key updates leak more information than those with backward uni-directional key updates, which further leaks more information than those with no-directional key updates.

Lemma 2. *For any sets $\mathcal{K}, \mathcal{T}, \mathcal{C}$ and any $\mathsf{cc} \in \{\mathsf{uni}, \mathsf{bi}\}$, we have $\mathcal{K}_{\mathsf{no}}^* \subseteq \mathcal{K}_{\mathsf{b\text{-}uni}}^* \subseteq \mathcal{K}_{\mathsf{bi}}^*$, $\mathcal{T}_{\mathsf{no}}^* \subseteq \mathcal{T}_{\mathsf{b\text{-}uni}}^* \subseteq \mathcal{T}_{\mathsf{bi}}^*$, $\mathcal{C}_{\mathsf{no,cc}}^* \subseteq \mathcal{C}_{\mathsf{b\text{-}uni,cc}}^* \subseteq \mathcal{C}_{\mathsf{bi,cc}}^*$, $\tilde{\mathcal{L}}_{\mathsf{no,cc}}^* \subseteq \tilde{\mathcal{L}}_{\mathsf{b\text{-}uni,cc}}^* \subseteq \tilde{\mathcal{L}}_{\mathsf{bi,cc}}^*$, $\tilde{\mathcal{Q}}_{\mathsf{no,cc}}^* \subseteq \tilde{\mathcal{Q}}_{\mathsf{b\text{-}uni,cc}}^* \subseteq \tilde{\mathcal{Q}}_{\mathsf{bi,cc}}^*$, $\mathcal{L}_{\mathsf{no,cc}}^* \subseteq \mathcal{L}_{\mathsf{b\text{-}uni,cc}}^* \subseteq \mathcal{L}_{\mathsf{bi,cc}}^*$ and $\mathcal{Q}_{\mathsf{no,cc}}^* \subseteq \mathcal{Q}_{\mathsf{b\text{-}uni,cc}}^* \subseteq \mathcal{Q}_{\mathsf{bi,cc}}^*$.*

Proof. For any $\mathsf{cc} \in \{\mathsf{uni}, \mathsf{bi}\}$, the adversary infers more information in the bi-directional key update setting than in the backward uni-directional key update setting. For any $\mathcal{K}, \mathcal{T}, \mathcal{C}$, the inferred leakage sets $\mathcal{K}_{\mathsf{b\text{-}uni}}^*$ and $\mathcal{K}_{\mathsf{bi}}^*$ are computed by Eq. (1),(2), then we have $\mathcal{K}_{\mathsf{no}}^* \subseteq \mathcal{K}_{\mathsf{b\text{-}uni}}^* \subseteq \mathcal{K}_{\mathsf{bi}}^*$. By Eq. (4), (5), (6), we have $\mathcal{T}_{\mathsf{no}}^* \subseteq \mathcal{T}_{\mathsf{b\text{-}uni}}^* \subseteq \mathcal{T}_{\mathsf{bi}}^*$ and $\mathcal{C}_{\mathsf{no,cc}}^* \subseteq \mathcal{C}_{\mathsf{b\text{-}uni,cc}}^* \subseteq \mathcal{C}_{\mathsf{bi,cc}}^*$. Then we obtain $\tilde{\mathcal{L}}_{\mathsf{no,cc}}^* \subseteq \tilde{\mathcal{L}}_{\mathsf{b\text{-}uni,cc}}^* \subseteq \tilde{\mathcal{L}}_{\mathsf{bi,cc}}^*$ and $\tilde{\mathcal{Q}}_{\mathsf{no,cc}}^* \subseteq \tilde{\mathcal{Q}}_{\mathsf{b\text{-}uni,cc}}^* \subseteq \tilde{\mathcal{Q}}_{\mathsf{bi,cc}}^*$ by Remark 1. We compute $\mathcal{L}^*$ and $\mathcal{Q}^*$ from $\mathcal{L}$ and $\mathcal{Q}$ with $\mathcal{T}^*$, and then we have $\mathcal{L}_{\mathsf{no,cc}}^* \subseteq \mathcal{L}_{\mathsf{b\text{-}uni,cc}}^* \subseteq \mathcal{L}_{\mathsf{bi,cc}}^*$ and $\mathcal{Q}_{\mathsf{no,cc}}^* \subseteq \mathcal{Q}_{\mathsf{b\text{-}uni,cc}}^* \subseteq \mathcal{Q}_{\mathsf{bi,cc}}^*$, which follows from $\mathcal{T}_{\mathsf{no}}^* \subseteq \mathcal{T}_{\mathsf{b\text{-}uni}}^* \subseteq \mathcal{T}_{\mathsf{bi}}^*$. □

Lemma 3. *For any sets $\mathcal{K}, \mathcal{T}, \mathcal{C}$ and any $\mathsf{kk} \in \{\mathsf{b\text{-}uni}, \mathsf{bi}\}$, we have $\mathcal{C}_{\mathsf{kk,uni}}^* \subseteq \mathcal{C}_{\mathsf{kk,bi}}^*$, $\tilde{\mathcal{L}}_{\mathsf{kk,uni}}^* \subseteq \tilde{\mathcal{L}}_{\mathsf{kk,bi}}^*$, $\tilde{\mathcal{Q}}_{\mathsf{kk,uni}}^* \subseteq \tilde{\mathcal{Q}}_{\mathsf{kk,bi}}^*$, $\mathcal{L}_{\mathsf{kk,uni}}^* \subseteq \mathcal{L}_{\mathsf{kk,bi}}^*$ and $\mathcal{Q}_{\mathsf{kk,uni}}^* \subseteq \mathcal{Q}_{\mathsf{kk,bi}}^*$.*

Proof. This follows similarly as in Lemma 2 and thus we omit the details. □

(**bi,bi**)-**notion** $\Longleftrightarrow$ (**b-uni,bi**)-**notion**. We prove the equivalence of $(\mathsf{bi}, \mathsf{bi})$-variant and the $(\mathsf{b\text{-}uni}, \mathsf{bi})$-variant in Theorem 1, which is based on the equivalence of trivial win conditions in Lemmas 4, 5, 6 and 7. We compare the relations of trivial win conditions in the two settings one by one (see Sect. 2.2).

Lemma 4. *For any sets $\mathcal{K}, \mathcal{T}, \mathcal{C}$, we have $\mathcal{K}_{\mathsf{bi}}^* \cap \mathcal{C}_{\mathsf{bi,bi}}^* \neq \emptyset \Longleftrightarrow \mathcal{K}_{\mathsf{b\text{-}uni}}^* \cap \mathcal{C}_{\mathsf{b\text{-}uni,bi}}^* \neq \emptyset$.*

Proof. From Lemma 2, we know that $\mathcal{K}_{\mathsf{b\text{-}uni}}^* \subseteq \mathcal{K}_{\mathsf{bi}}^*$ and $\mathcal{C}_{\mathsf{b\text{-}uni,bi}}^* \subseteq \mathcal{C}_{\mathsf{bi,bi}}^*$, so $\mathcal{K}_{\mathsf{b\text{-}uni}}^* \cap \mathcal{C}_{\mathsf{b\text{-}uni,bi}}^* \subseteq \mathcal{K}_{\mathsf{bi}}^* \cap \mathcal{C}_{\mathsf{bi,bi}}^*$. It is sufficient to prove $\mathcal{K}_{\mathsf{bi}}^* \cap \mathcal{C}_{\mathsf{bi,bi}}^* \neq \emptyset \Rightarrow \mathcal{K}_{\mathsf{b\text{-}uni}}^* \cap \mathcal{C}_{\mathsf{b\text{-}uni,bi}}^* \neq \emptyset$. If the adversary never queries any challenge-equal ciphertext in an epoch in $\{0, \ldots, l\} \setminus \mathcal{IR}$, it will not obtain any challenge-equal ciphertext in this set even in the bi-directional ciphertext update setting by Eq. (6), i.e., $\mathcal{C}_{\mathsf{bi,bi}}^* \cap \{0, \ldots, l\} \setminus \mathcal{IR} = \emptyset$. This contradicts $\mathcal{K}_{\mathsf{bi}}^* \cap \mathcal{C}_{\mathsf{bi,bi}}^* \neq \emptyset$, since $\mathcal{K}_{\mathsf{bi}}^* = \{0, \ldots, l\} \setminus \mathcal{IR}$ by Lemma 1. There exists an epoch $\mathsf{e}' \in \{0, \ldots, l\} \setminus \mathcal{IR}$, in which the adversary queries a challenge-equal ciphertext, i.e., $\mathsf{e}' \in \mathcal{C} \cap \mathcal{K}_{\mathsf{bi}}^*$.

Note that $\mathcal{K}_{\mathsf{bi}}^* = \mathcal{K}_{\mathsf{b\text{-}uni}}^* \cup \mathcal{K}_{\mathsf{f\text{-}uni}}^*$. If $\mathsf{e}' \in \mathcal{K}_{\mathsf{b\text{-}uni}}^*$, then $\mathsf{e}' \in \mathcal{K}_{\mathsf{b\text{-}uni}}^* \cup \mathcal{C} \subseteq \mathcal{K}_{\mathsf{b\text{-}uni}}^* \cup \mathcal{C}_{\mathsf{b\text{-}uni,bi}}^*$, so $\mathcal{K}_{\mathsf{b\text{-}uni}}^* \cup \mathcal{C}_{\mathsf{b\text{-}uni,bi}}^* \neq \emptyset$. If $\mathsf{e}' \in \mathcal{K}_{\mathsf{f\text{-}uni}}^*$, then there exists a smaller epoch e'' than e' such that $\mathsf{e}'' \in \mathcal{K}$ and the set $\{\mathsf{e}'', \ldots, \mathsf{e}'\} \subseteq \mathcal{T}$ by Remark 2. Hence, the adversary can degrade the message from e' to e'' to know $\tilde{c}_{\mathsf{e}''}$ in the bi-directional ciphertext update setting. Therefore, we have $\mathsf{e}'' \in \mathcal{K} \cap \mathcal{C}_{\mathsf{b\text{-}uni,bi}}^* \subseteq \mathcal{K}_{\mathsf{b\text{-}uni}}^* \cap \mathcal{C}_{\mathsf{b\text{-}uni,bi}}^*$, so $\mathcal{K}_{\mathsf{b\text{-}uni}}^* \cup \mathcal{C}_{\mathsf{b\text{-}uni,bi}}^* \neq \emptyset$. □

Lemma 5. *For any set $\mathcal{K}, \mathcal{T}, \mathcal{C}$, suppose $\mathcal{K}_{\mathsf{kk}}^* \cap \mathcal{C}_{\mathsf{kk,bi}}^* = \emptyset$ for $\mathsf{kk} \in \{\mathsf{bi}, \mathsf{b\text{-}uni}\}$, then $\tilde{\mathsf{e}} \in \mathcal{T}_{\mathsf{bi}}^* \Longleftrightarrow \tilde{\mathsf{e}} \in \mathcal{T}_{\mathsf{b\text{-}uni}}^*$.*

Proof. From Lemma 2, we know that $\mathcal{T}_{\mathsf{b\text{-}uni}}^* \subseteq \mathcal{T}_{\mathsf{bi}}^*$, so if $\tilde{\mathsf{e}} \in \mathcal{T}_{\mathsf{b\text{-}uni}}^*$, then $\tilde{\mathsf{e}} \in \mathcal{T}_{\mathsf{bi}}^*$. It is sufficient to prove $\tilde{\mathsf{e}} \in \mathcal{T}_{\mathsf{bi}}^* \Rightarrow \tilde{\mathsf{e}} \in \mathcal{T}_{\mathsf{b\text{-}uni}}^*$. Because the adversary queries the challenge ciphertext in epoch $\tilde{\mathsf{e}}$ (i.e., $\tilde{\mathsf{e}} \in \mathcal{C} \subseteq \mathcal{C}_{\mathsf{bi,bi}}^*$) and $\mathcal{K}_{\mathsf{bi}}^* \cap \mathcal{C}_{\mathsf{bi,bi}}^* = \emptyset$, we have

$\tilde{e} \notin \mathcal{K}_{bi}^*$. $\Delta_{\tilde{e}}$ cannot be inferred from the successive keys in epochs $\tilde{e} - 1$ and $\tilde{e}$. Therefore, if $\tilde{e} \in \mathcal{T}_{bi}^*$, it must be obtained via corrupting, that is $\tilde{e} \in \mathcal{T}$. Since $\mathcal{T} \subseteq \mathcal{T}_{b\text{-uni}}^*$, we have $\tilde{e} \in \mathcal{T}_{b\text{-uni}}^*$. □

Remark 3. Note that $\mathcal{O}.\mathsf{Upd}(\bar{c})$ is queried or not is independent of the direction of key and ciphertext updates. Thus it will be the same whether this trivial win condition is triggered or not in all variants.

Lemma 6. *For any set* $\mathcal{K}, \mathcal{T}, \mathcal{C}$, *suppose* $\mathcal{K}_{kk}^* \cap \mathcal{C}_{kk,bi}^* = \emptyset$ *for* $kk \in \{bi, b\text{-uni}\}$, *then* $(c, e) \in \tilde{\mathcal{L}}_{bi,bi}^* \Longleftrightarrow (c, e) \in \tilde{\mathcal{L}}_{b\text{-uni},bi}^*$.

Proof. By Remark 1 and Lemma 2, we know that $(c, e) \in \tilde{\mathcal{L}} \Longleftrightarrow e \in \mathcal{C}^*$ and $\mathcal{C}_{b\text{-uni},bi}^* \subseteq \mathcal{C}_{bi,bi}^*$. So if $(c, e) \in \mathcal{L}_{b\text{-uni},bi}^*$, then $e \in \mathcal{C}_{b\text{-uni},bi}^* \subseteq \mathcal{C}_{bi,bi}^*$. Thus, we have $(c, e) \in \mathcal{L}_{bi,bi}^*$.

If $(c, e) \in \tilde{\mathcal{L}}_{bi,bi}^*$, that is $e \in \mathcal{C}_{bi,bi}^*$, then we know $e \in \mathcal{IR}$ by the assumption $\mathcal{K}_{bi}^* \cap \mathcal{C}_{bi,bi}^* = \emptyset$ and the fact that $\mathcal{K}_{bi}^* = \{0, \dots, l\} \setminus \mathcal{IR}$ from Lemma 1. Suppose $\{fwl, \dots, e\}$ is the last insulated region. If the adversary never queries the challenge-equal ciphertext in the epoch in this set, then it cannot infer any challenge-equal ciphertext in epoch e, which contradicts $e \in \mathcal{C}_{bi,bi}^*$. Therefore we assume the adversary queries a challenge-equal ciphertext in epoch e', where $e' \in \{fwl, \dots, e\}$. Since $\{fwl, \dots, e\} \subseteq \mathcal{T}$ even in the backward uni-directional update setting, the adversary can update challenge-equal ciphertext from epoch e' to e, i.e., $e \in \mathcal{C}^*$. So $(c, e) \in \tilde{\mathcal{L}}_{b\text{-uni},bi}^*$. □

Lemma 7. *For any set* $\mathcal{K}, \mathcal{T}, \mathcal{C}$, *suppose* $\mathcal{K}_{kk}^* \cap \mathcal{C}_{kk,bi}^* = \emptyset$ *for* $kk \in \{bi, b\text{-uni}\}$, *then* $(m', e) \in \tilde{\mathcal{Q}}_{bi,bi}^* \Longleftrightarrow (m', e) \in \tilde{\mathcal{Q}}_{b\text{-uni},bi}^*$.

Proof. By Remark 2, we know that $(m', e) \in \tilde{\mathcal{Q}} \Longleftrightarrow e \in \mathcal{C}^*$. And the rest of the proof is similar to that of Lemma 6. □

Theorem 1. *Let* UE *be an updatable encryption scheme and confidentiality* notion $\in \{\mathsf{detIND\text{-}UE\text{-}CPA}, \mathsf{randIND\text{-}UE\text{-}CPA}, \mathsf{detIND\text{-}UE\text{-}CCA}, \mathsf{randIND\text{-}UE\text{-}CCA}\}$. *For any* (bi, bi)-notion *adversary* $\mathcal{A}$ *against* UE, *there exists a* $(b\text{-uni}, bi)$-notion *adversary* $\mathcal{B}_1$ *against* UE *such that*

$$\mathsf{Adv}_{\mathsf{UE},\mathcal{A}}^{(bi,bi)\text{-notion}}(1^\lambda) = \mathsf{Adv}_{\mathsf{UE},\mathcal{B}_1}^{(b\text{-uni},bi)\text{-notion}}(1^\lambda).$$

Proof. We construct a reduction $\mathcal{B}_1$ who runs the $(b\text{-uni}, bi)$-notion experiment and simulates all responses of the queries made by (bi, bi)-notion adversary $\mathcal{A}$. The reduction $\mathcal{B}_1$ works by sending all the queries of $\mathcal{A}$ to its own challenger and returning the responses to $\mathcal{A}$. At last, $\mathcal{B}_1$ sends the guessing result received from $\mathcal{A}$ to its own challenger. The challenger will check whether the reduction wins or not. If the reduction triggers the trivial win conditions, it will lose the game. The reduction also forwards the experiment result to $\mathcal{A}$.

Notice that Lemmas 4, 5, 6, 7, and Remark 3 exactly include all the trivial win conditions in the confidentiality game (see Fig. 5). Thus, we can conclude that the trivial win conditions in the (bi, bi)-notion and $(b\text{-uni}, bi)$-notion

games are equivalent. If no trivial conditions are triggered by $\mathcal{A}$, there will be no trivial win conditions triggered by $\mathcal{B}$. But if a condition is triggered in the $(\mathsf{bi}, \mathsf{bi})$-notion, the same condition will also be triggered in the $(\mathsf{b\text{-}uni}, \mathsf{bi})$-notion. Therefore, the reduction perfectly simulates the $(\mathsf{bi}, \mathsf{bi})$-notion game to $\mathcal{A}$. Then $\mathsf{Adv}_{\mathsf{UE},\mathcal{A}}^{(\mathsf{bi},\mathsf{bi})\text{-notion}}(1^\lambda) = \mathsf{Adv}_{\mathsf{UE},\mathcal{B}_1}^{(\mathsf{b\text{-}uni},\mathsf{bi})\text{-notion}}(1^\lambda).$ □

$(\mathbf{b\text{-}uni},\mathbf{uni})\text{-}\mathbf{notion} \iff (\mathbf{no},\mathbf{uni})\text{-}\mathbf{notion}$. We further prove the equivalence of $(\mathsf{b\text{-}uni}, \mathsf{uni})$-variant and the $(\mathsf{b\text{-}uni}, \mathsf{uni})$-variant in Theorem 2, which is based on the equivalence of trivial win conditions in Lemmas 8, 9, 10 and 11.

Lemma 8. *For any set $\mathcal{K}, \mathcal{T}, \mathcal{C}$, we have $\mathcal{K}_{\mathsf{b\text{-}uni}}^* \cap \mathcal{C}_{\mathsf{b\text{-}uni},\mathsf{uni}}^* \neq \emptyset \iff \mathcal{K}_{\mathsf{no}}^* \cap \mathcal{C}_{\mathsf{no},\mathsf{uni}}^* \neq \emptyset$.*

Proof. From Lemma 2, we know that $\mathcal{K}_{\mathsf{no}}^* \subseteq \mathcal{K}_{\mathsf{b\text{-}uni}}^*$ and $\mathcal{C}_{\mathsf{no},\mathsf{uni}}^* \subseteq \mathcal{C}_{\mathsf{b\text{-}uni},\mathsf{uni}}^*$, so $\mathcal{K}_{\mathsf{no}}^* \cap \mathcal{C}_{\mathsf{no},\mathsf{uni}}^* \subseteq \mathcal{K}_{\mathsf{b\text{-}uni}}^* \cap \mathcal{C}_{\mathsf{b\text{-}uni},\mathsf{uni}}^*$. It is sufficient to prove $\mathcal{K}_{\mathsf{b\text{-}uni}}^* \cap \mathcal{C}_{\mathsf{b\text{-}uni},\mathsf{uni}}^* \neq \emptyset \Rightarrow \mathcal{K}_{\mathsf{no}}^* \cap \mathcal{C}_{\mathsf{no},\mathsf{bi}}^* \neq \emptyset$.

Suppose there exists an epoch $\mathsf{e} \in \mathcal{K}_{\mathsf{b\text{-}uni}}^* \cap \mathcal{C}_{\mathsf{b\text{-}uni},\mathsf{uni}}^*$. From $\mathsf{e} \in \mathcal{K}_{\mathsf{b\text{-}uni}}^*$ and Remark 2, there is an epoch e_b after e, satisfying $\mathsf{e}_\mathsf{b} \in \mathcal{K}$ and $\{\mathsf{e}, \ldots, \mathsf{e}_\mathsf{b}\} \in \mathcal{T}$. From $\mathsf{e} \in \mathcal{C}_{\mathsf{b\text{-}uni},\mathsf{uni}}^*$ and the definition of $\mathcal{C}_{\mathsf{b\text{-}uni},\mathsf{uni}}^*$ in Eq. (5), we know that there exists an epoch e_c before e such that the adversary asks for the challenge ciphertext in epoch e_c (i.e., $\mathsf{e}_\mathsf{c} \in \mathcal{C}$) and $\{\mathsf{e}_\mathsf{c}, \ldots, \mathsf{e}\} \in \mathcal{T}_{\mathsf{b\text{-}uni}}^*$. If the set $\{\mathsf{e}_\mathsf{c}, \ldots, \mathsf{e}\} \subseteq \mathcal{T}$, then we can upgrade the ciphertext from epoch e_c to epoch e_b even in the no-directional key update setting, since $\mathsf{e}_\mathsf{c} \in \mathcal{C}$ and $\{\mathsf{e}_\mathsf{c}, \ldots, \mathsf{e}, \ldots, \mathsf{e}_\mathsf{b}\} \in \mathcal{T}$. Therefore, we have $\mathsf{e}_\mathsf{b} \in \mathcal{K}_{\mathsf{no}}^* \cap \mathcal{C}_{\mathsf{no},\mathsf{bi}}^*$.

If not every epoch in the set $\{\mathsf{e}_\mathsf{c}, \ldots, \mathsf{e}\}$ is in $\mathcal{T}$ (i.e., there is an epoch $\mathsf{e}_\mathsf{s} \in \mathcal{T}_{\mathsf{b\text{-}uni}}^* \setminus \mathcal{T}$), then by Eq. (4), we know that $\mathsf{e}_\mathsf{s} - 1$ and e_s are in $\mathcal{K}_{\mathsf{b\text{-}uni}}^*$. Moreover, we have $\mathsf{e}_\mathsf{s} - 1 \in \mathcal{K}$, because the epoch key in $\mathsf{e}_\mathsf{s} - 1$ cannot be inferred from the key in $\mathsf{e}_\mathsf{s} - 1$ in the backward uni-directional key update setting as $\mathsf{e}_\mathsf{s} \notin \mathcal{T}$ and the adversary can only learn the epoch key in $\mathsf{e}_\mathsf{s} - 1$ from querying the corruption oracle. If $\{\mathsf{e}_\mathsf{c}, \ldots, \mathsf{e}_\mathsf{s} - 1\} \in \mathcal{T}$, we can upgrade ciphertexts from epoch e_c to epoch $\mathsf{e}_\mathsf{s} - 1$ even in the no-directional key update setting, that is $\mathsf{e}_\mathsf{s} - 1 \in \mathcal{K}_{\mathsf{no}}^* \cap \mathcal{C}_{\mathsf{no},\mathsf{bi}}^*$. Otherwise, we repeat this step, substitute $\mathsf{e}_\mathsf{s} - 1$ with a smaller epoch $\mathsf{e}_\mathsf{s} - j$ in the next iteration for some $j > 1$ such that $\mathsf{e}_\mathsf{s} - j \in \mathcal{K}$ and $\{\mathsf{e}_\mathsf{c}, \ldots, \mathsf{e}_\mathsf{s} - j\} \in \mathcal{T}_{\mathsf{b\text{-}uni}}^*$, and check if all epochs in $\{\mathsf{e}_\mathsf{c}, \ldots, \mathsf{e}_\mathsf{s} - j\}$ are in $\mathcal{T}$. Since the epoch length is limited, we will stop at an epoch, say $\mathsf{e}_\mathsf{s} - k$, for some $k > 1$ such that $\mathsf{e}_\mathsf{s} - k \in \mathcal{K}$ and $\{\mathsf{e}_\mathsf{c}, \ldots, \mathsf{e}_\mathsf{s} - k\} \in \mathcal{T}$. We can upgrade ciphertext from epoch e_c to epoch $\mathsf{e}_\mathsf{s} - k$ even in the no-directional key update setting, that is $\mathsf{e}_\mathsf{s} - k \in \mathcal{K}_{\mathsf{no}}^* \cap \mathcal{C}_{\mathsf{no},\mathsf{bi}}^*$, so $\mathcal{K}_{\mathsf{no}}^* \cap \mathcal{C}_{\mathsf{no},\mathsf{bi}}^* \neq \emptyset$. □

Lemma 9. *For any set $\mathcal{K}, \mathcal{T}, \mathcal{C}$, suppose $\mathcal{K}_{\mathsf{kk}}^* \cap \mathcal{C}_{\mathsf{kk},\mathsf{uni}}^* = \emptyset$ for $\mathsf{kk} \in \{\mathsf{b\text{-}uni}, \mathsf{no}\}$, then $\tilde{\mathsf{e}} \in \mathcal{T}_{\mathsf{b\text{-}uni}}^* \iff \tilde{\mathsf{e}} \in \mathcal{T}_{\mathsf{no}}^*$.*

Proof. The proof is similar to that of Lemma 5. We provide it in Appendix D.1. □

Lemma 10. *For any set $\mathcal{K}, \mathcal{T}, \mathcal{C}$, suppose $\mathcal{K}_{\mathsf{kk}}^* \cap \mathcal{C}_{\mathsf{kk},\mathsf{uni}}^* = \emptyset$ for $\mathsf{kk} \in \{\mathsf{b\text{-}uni}, \mathsf{no}\}$, then $(\mathsf{c}, \mathsf{e}) \in \tilde{\mathcal{L}}_{\mathsf{b\text{-}uni}}^* \iff (\mathsf{c}, \mathsf{e}) \in \tilde{\mathcal{L}}_{\mathsf{no}}^*$.*

Proof. The proof is similar to that of Lemma 6. We provide it in Appendix D.2. □

Lemma 11. *For any set* $\mathcal{K}, \mathcal{T}, \mathcal{C}$, *suppose* $\mathcal{K}_{kk}^* \cap \mathcal{C}_{kk,uni}^* = \emptyset$ *for* kk $\in$ {b-uni, no}, *then* $(m', e) \in \tilde{\mathcal{Q}}_{b\text{-uni,uni}}^* \Longleftrightarrow (m', e) \in \tilde{\mathcal{Q}}_{no,uni}^*$.

Proof. By Remark 1, we know that $(m', e) \in \tilde{\mathcal{Q}} \Longleftrightarrow e \in \mathcal{C}^*$. The rest of the proof is similar to that of Lemma 10. □

Theorem 2. *Let* UE *be an updatable encryption scheme and confidentiality* notion $\in$ {detIND-UE-CPA, randIND-UE-CPA, detIND-UE-CCA, randIND-UE-CCA}. *For any* (b-uni, uni)-notion *adversary* $\mathcal{A}$ *against* UE, *there exists a* (no, uni)-notion *adversary* $\mathcal{B}_2$ *against* UE *such that*

$$\mathsf{Adv}_{\mathsf{UE},\mathcal{A}}^{(b\text{-uni,uni})\text{-notion}}(1^\lambda) = \mathsf{Adv}_{\mathsf{UE},\mathcal{B}_2}^{(no,uni)\text{-notion}}(1^\lambda)$$

Proof. The proof is similar to that of Theorem 1. We provide it in Appendix D.3. □

Theorem 3. *For* notion $\in$ {detIND-UE-CPA, randIND-UE-CPA, detIND-UE-CCA, randIND-UE-CCA}, *Fig. 6 is the relations among the eight variants on the same confidentiality notion.*

Proof. We conclude the relations among the eight variants on confidentiality from Theorems 1 and 2, together with the previous conclusions in [10], which proved that a UE scheme with bi-directional key updates is equivalent to the one with forward-leak uni-directional key updates, shown in Fig. 7.
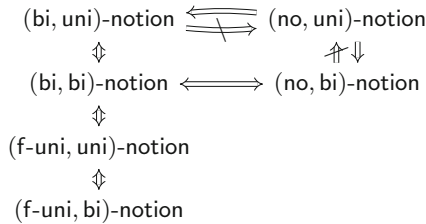

Fig. 7. Relations among the six variants of confidentiality for notion $\in$ {detIND-UE-CPA, randIND-UE-CPA, detIND-UE-CCA, randIND-UE-CCA} [10].

Thus, we have the relations among the eight variants on confidentiality in Fig. 6 by the equivalences: (bi, bi)-notion $\Longleftrightarrow$ (b-uni, bi)-notion from Theorem 1 and (b-uni, uni)-notion $\Longleftrightarrow$ (no, uni)-notion from Theorem 2 and Fig. 7. □

3.2   Relations Among Integrity Notions

The relations of the eight variants on integrity are illustrated in Fig. 8. We first
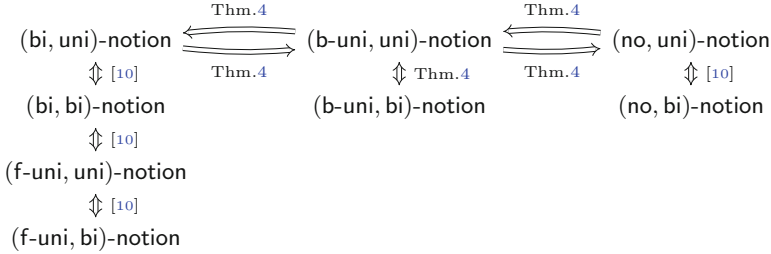prove two equivalence of trivial win conditions in Lemmas 12, 14 and 16.

$$
\begin{array}{ccccc}
 & \text{Thm.4} & & \text{Thm.4} & \\
\text{(bi, uni)-notion} & \xLeftrightarrow{} & \text{(b-uni, uni)-notion} & \xLeftrightarrow{} & \text{(no, uni)-notion} \\
\Updownarrow [10] & \text{Thm.4} & \Updownarrow \text{Thm.4} & \text{Thm.4} & \Updownarrow [10] \\
\text{(bi, bi)-notion} & & \text{(b-uni, bi)-notion} & & \text{(no, bi)-notion} \\
\Updownarrow [10] & & & & \\
\text{(f-uni, uni)-notion} & & & & \\
\Updownarrow [10] & & & & \\
\text{(f-uni, bi)-notion} & & & &
\end{array}
$$

Fig. 8. Relations among the eight variants of integrity for notion $\in$ {IND-CTXT,
IND-PTXT}.

Lemma 12. *For any set* $\mathcal{K}, \mathcal{T}, \mathcal{C}$, *we have* $\mathsf{e} \in \hat{\mathcal{K}}^*_{\text{b-uni}} \iff \mathsf{e} \in \hat{\mathcal{K}}^*_{\text{kk}}$ *for* $\text{kk} \in$
{f-uni, bi, no}.

Proof. From Eq. (3), we know that the computation of the extended set $\hat{\mathcal{K}}^*$ is
independent of the direction of key updates.   □

Lemma 13 ([10], **Lemma 3.11**). *For any set* $\mathcal{K}, \mathcal{T}, \mathcal{C}$, *suppose* $\mathsf{e} \notin \hat{\mathcal{K}}^*$, *then*
we have $(\mathsf{c}, \mathsf{e}) \in \mathcal{L}^*_{\text{kk,cc}} \iff (\mathsf{c}, \mathsf{e}) \in \mathcal{L}^*_{\text{kk}',\text{cc}'}$ *for any* $\text{kk}, \text{kk}' \in$ {f-uni, bi, no} *and*
$\text{cc}, \text{cc}' \in$ {uni, bi}.

Lemma 14. *For any set* $\mathcal{K}, \mathcal{T}, \mathcal{C}$, *suppose* $\mathsf{e} \notin \hat{\mathcal{K}}^*$, *then* $(\mathsf{c}, \mathsf{e}) \in \mathcal{L}^*_{\text{b-uni,cc}} \iff$
$(\mathsf{c}, \mathsf{e}) \in \mathcal{L}^*_{\text{kk}',\text{cc}'}$ *for any* $\text{kk}' \in$ {f-uni, b-uni, bi, no} *and* $\text{cc}, \text{cc}' \in$ {uni, bi}.

Proof. It follows directly from Lemma 13 and $\mathcal{L}^*_{\text{no,cc}} \subseteq \mathcal{L}^*_{\text{b-uni,cc}} \subseteq \mathcal{L}^*_{\text{bi,cc}}$ by
Lemma 2 for any $\text{cc} \in$ {uni, bi}.   □

Lemma 15 ([10], **Lemma 3.12**). *For any set* $\mathcal{K}, \mathcal{T}, \mathcal{C}$, *suppose* $\mathsf{e} \notin \hat{\mathcal{K}}^*$, *then*
we have $(\mathsf{m}', \mathsf{e}) \in \tilde{\mathcal{Q}}^*_{\text{kk,cc}} \iff (\mathsf{m}', \mathsf{e}) \in \tilde{\mathcal{Q}}^*_{\text{kk}',\text{cc}'}$ *for any* $\text{kk}, \text{kk}' \in$ {f-uni, bi, no} *and*
$\text{cc}, \text{cc}' \in$ {uni, bi}.

Lemma 16. *For any set* $\mathcal{K}, \mathcal{T}, \mathcal{C}$, *suppose* $\mathsf{e} \notin \hat{\mathcal{K}}^*$, *then we have* $(\mathsf{m}', \mathsf{e}) \in$
$\tilde{\mathcal{Q}}^*_{\text{b-uni,cc}} \iff (\mathsf{m}', \mathsf{e}) \in \tilde{\mathcal{Q}}^*_{\text{kk}',\text{cc}'}$ *for any* $\text{kk}' \in$ {f-uni, b-uni, bi, no} *and* $\text{cc}, \text{cc}' \in$
{uni, bi}.

Proof. It follows directly from Lemma 15 and $\mathcal{Q}^*_{\text{no,cc}} \subseteq \mathcal{Q}^*_{\text{b-uni,cc}} \subseteq \mathcal{Q}^*_{\text{bi,cc}}$ by
Lemma 3 for any $\text{cc} \in$ {uni, bi}.   □

Theorem 4. *Let* UE *be an updatable encryption scheme, the integrity notion*
notion $\in$ {INT-CTXT, INT-PTXT}. *For any* (b-uni, cc)-notion *adversary* $\mathcal{A}$
against UE, *there exists a* (kk', cc')-notion *adversary* $\mathcal{B}_4$ *against* UE *such that*

$$\mathsf{Adv}^{(\text{b-uni,cc})\text{-notion}}_{\mathsf{UE}, \mathcal{A}}(1^\lambda) = \mathsf{Adv}^{(\text{kk}',\text{cc}')\text{-notion}}_{\mathsf{UE}, \mathcal{B}_4}(1^\lambda)$$

for any $\text{kk}' \in$ {f-uni, b-uni, bi, no} *and* $\text{cc}, \text{cc}' \in$ {uni, bi}.

Proof. The proof is similar to that of Theorem 1. We provide the proof in Appendix D.4. □

Theorem 5. *For* notion $\in \{$INT-CTXT, INT-PTXT$\}$, *Fig.* 8 *shows the relations among the eight variants on the same integrity notion.*

Proof. We conclude the relations from Theorem 4, together with the previous conclusions in [10] that $($kk, cc$)$-notion $\iff$ $($no, cc$')$-notion for notion $\in \{$IND-CTXT, IND-PTXT$\}$, kk $\in ($bi, f-uni$)$ and cc, cc$' \in ($bi, uni$)$. □

4 Conclusion

The relations among various security notions for UE should be clearly investigated before any valuable constructions. We provided a detailed comparison of every security notion in the four key update settings, and our results showed that the UE schemes in the no-directional key update setting, which were believed to be strictly stronger than others, are equivalent to those in the backward-leak uni-directional key update setting. As future work, we intend to develop an efficient UE scheme with backward-leak uni-directional key updates.

A Related Work

UE schemes can be built from various cryptographic primitives. The seminal UE scheme BLMR was proposed by [3] as an application of almost key homomorphic pseudorandom functions, which satisfies IND-ENC instead of IND-UPD. An ElGamal-based scheme RISE was introduced by [12] to achieve both security definitions. To provide integrity protection, Klooß et al. [11] constructed two generic schemes based on Encrypt-and-MAC and the Naor-Yung transform [13]. Boyd et al. [4] designed three IND-UE-CPA secure schemes, called SHINE, based on the random-looking permutation. Jiang [10] provided a quantum-resistant scheme based on the decisional LWE [15]. The first UE scheme with backward uni-directional key was presented in Nishimaki [14] based on the Regev PKE scheme, in which a scheme with no-directional key updates is also constructed based on one-way functions [8] and indistinguishability obfuscation [1]. Slamanig and Striecks presented a pairing backward uni-directional scheme and a pairing-based no-directional scheme from ciphertext puncturable encryption [9].

An independent work was concurrently proposed by [7]. The main difference between their work and ours is that we provide a detailed comparison of every security notion in every kind of UE. Their work gave a detailed proof for equivalence of confidentiality notion in the no-directional and backward uni-directional

key update setting for UE schemes with uni-directional ciphertext updates (i.e., $(\mathsf{b\text{-}uni}, \mathsf{uni})$-notion $\iff$ $(\mathsf{no}, \mathsf{uni})$-notion). Our proof for this equivalence is different, and we also provide a detailed proof for UE schemes with bi-directional ciphertext updates and also the equivalence among integrity notions.

B   An Example of Leakage Sets

An example is given in Fig. 9 to show how to compute leakage sets. We assume the adversary corrupts epoch keys in epochs in $\mathcal{K} = \{\mathsf{e} - 5, \mathsf{e} - 4, \mathsf{e} - 3, \mathsf{e} - 1\}$ and corrupts tokens in $\mathcal{T} = \{\mathsf{e} - 4, \mathsf{e} - 3, \mathsf{e} - 1, \mathsf{e}\}$, and queries a non-challenge ciphertext, say $\mathbf{c}_{\mathsf{e}-5}$, in epoch $\mathsf{e} - 5$.

In the no-directional key update setting, the adversary cannot infer extra keys and tokens, i.e., $\mathcal{K}_{\mathsf{no}}^* = \mathcal{K}$ and $\mathcal{T}_{\mathsf{no}}^* = \mathcal{T}$. However, it can infer ciphertexts in epoch $\mathsf{e} - 4, \mathsf{e} - 3$ by using $\mathbf{c}_{\mathsf{e}-5}$ and tokens in epochs $\mathsf{e} - 4$ and $\mathsf{e} - 3$, but cannot infer the ciphertexts in epochs from $\mathsf{e} - 2$ to e, because the token in epoch $\mathsf{e} - 2$ is unknown to the adversary in the no-directional key update setting.

Epoch		$\mathsf{e} - 5$	$\mathsf{e} - 4$	$\mathsf{e} - 3$	$\mathsf{e} - 2$	$\mathsf{e} - 1$	e
$\mathcal{K}$		✓	✓	✓	×	✓	×
$\mathcal{T}$	×	✓	✓	×	✓	✓	
$\mathcal{K}_{\mathsf{no}}^*$		✓	✓	✓	×	✓	×
$\mathcal{T}_{\mathsf{no}}^*$	×	✓	✓	×	✓	✓	
$\mathcal{K}_{\mathsf{b\text{-}uni}}^*$		✓	✓	✓	✓	✓	×
$\mathcal{T}_{\mathsf{b\text{-}uni}}^*$	×	✓	✓	✓	✓	✓	
$\mathcal{K}_{\mathsf{f\text{-}uni}}^*$		✓	✓	✓	×	✓	✓
$\mathcal{T}_{\mathsf{f\text{-}uni}}^*$	×	✓	✓	×	✓	✓	

Fig. 9. Example of leakage sets. Marks ✓ and × indicate if an epoch key or epoch token is corrupted. The green mark ✓ indicates an epoch key or epoch token can be inferred from other corrupted keys and tokens. (Color figure online)

In the backward uni-directional key update setting, the adversary can infer the key in epoch $\mathsf{e} - 2$ from the known token and key in epoch $\mathsf{e} - 1$, and further infer the token in the epoch $\mathsf{e} - 2$, since the key in epoch $\mathsf{e} - 3$ is also corrupted, i.e., $\mathcal{K}_{\mathsf{b\text{-}uni}}^* = \{\mathsf{e} - 5, \ldots, \mathsf{e} - 1\}$ and $\mathcal{T}_{\mathsf{b\text{-}uni}}^* = \{\mathsf{e} - 4, \ldots, \mathsf{e}\}$. Moreover, it can infer the ciphertexts in epoch from $\mathsf{e} - 4$ to e by $\mathbf{c}_{\mathsf{e}-5}$ and $\mathcal{T}_{\mathsf{b\text{-}uni}}^*$.

In the forward uni-directional key update setting, the adversary cannot infer the token in epoch $\mathsf{e} - 2$, since the key in epoch $\mathsf{e} - 2$ is unknown to it. But it can infer the key in epoch e via the known key in $\mathsf{e} - 1$ and the known token in e, i.e., $\mathcal{K}_{\mathsf{f\text{-}uni}}^* = \mathcal{K} \cup \{\mathsf{e}\}$ and $\mathcal{T}_{\mathsf{f\text{-}uni}}^* = \mathcal{K}^*$. The ciphertext it can learn is the same as that in the no-directional key update setting.

C   Trivial Win Conditions

We give a detailed explanation for the trivial win conditions in each security game. If $\mathcal{K}^* \cap \mathcal{C}^* \neq \emptyset$, then there is an epoch i that the adversary knows the epoch key k_i and a challenge-equal ciphertext c_i in the same epoch. Then the adversary can decrypt the challenge-equal ciphertext c_i with its known key k_i and get the underlying message of c_i. Thus, it can trivially win the game by comparing the underlying message of c_i with the challenge input message $\bar{m}$. This condition should be checked in all confidentiality games.

For deterministic UE schemes, if $\tilde{e} \in \mathcal{T}^*$ or $\mathcal{O}.\mathsf{Upd}(\bar{c})$ is queried, then the adversary can obtain the updated ciphertext c_1 of the challenge input ciphertext $\bar{c}$, and therefore trivially win the game by comparing c_1 with the ciphertext it receives from its challenger.

In the CCA attack, the adversary has the access to the decryption oracle. For deterministic UE schemes, if $(c, e) \in \tilde{\mathcal{L}}^*$, the adversary can query the decryption oracle on the challenge-equal ciphertext c and receive its underlying message. For a randomized UE, it should also be prohibited if the decryption returns a message m' such that $m' = m_0$ or m_1, which is checked by $(m', e) \in \tilde{\mathcal{Q}}^*$.

If the adversary knows a ciphertext c_{e_0} in epoch e_0 and all tokens from epoch e_0 to e, it can forge a valid ciphertext in epoch e by updating c_{e_0} via the tokens from e_0 to e. It should be checked in both integrity games if $e \in \hat{\mathcal{K}}^*$ which is defined as Eq. (3). The challenger should also check if $\tilde{c}$ is a new ciphertext in the CTXT game, and if the adversary knows a ciphertext of m' in the CTXT game.

D   Proofs

D.1   Proof of Lemma 9

From Lemma 2, we know that $\mathcal{T}_{no}^* \subseteq \mathcal{T}_{b\text{-uni}}^*$, so if $\tilde{e} \in \mathcal{T}_{no}^*$, then $\tilde{e} \in \mathcal{T}_{b\text{-uni}}^*$. Notice that $\tilde{e} \notin \mathcal{K}_{b\text{-uni}}^*$, because the adversary queries the challenge ciphertext in the epoch $\tilde{e}$ and $\mathcal{K}_{b\text{-uni}}^* \cap \mathcal{C}_{b\text{-uni,bi}}^* = \emptyset$. Then $\Delta_{\tilde{e}}$ cannot be inferred from the successive keys in epochs $\tilde{e} - 1$ and $\tilde{e}$. Therefore, if $\tilde{e} \in \mathcal{T}_{b\text{-uni}}^*$, then it must be obtained from corrupting, that is $\tilde{e} \in \mathcal{T}$. Since $\mathcal{T} = \mathcal{T}_{no}^*$, we have $\tilde{e} \in \mathcal{T}_{no}^*$.

D.2   Proof of Lemma 10

By Remark 1 and Lemma 2 , we know that $(c, e) \in \tilde{\mathcal{L}} \iff e \in \mathcal{C}^*$ and $\mathcal{C}_{no,uni}^* \subseteq \mathcal{C}_{b\text{-uni,uni}}^*$. So if $(c, e) \in \mathcal{L}_{no,uni}^*$, then $e \in \mathcal{C}_{no,uni}^* \subseteq \mathcal{C}_{b\text{-uni,uni}}^*$. Thus, we have $(c, e) \in \tilde{\mathcal{L}}_{b\text{-uni}}^*$.

If $(c, e) \in \tilde{\mathcal{L}}_{b\text{-uni}}^*$, then $e \in \mathcal{C}_{b\text{-uni,uni}}^* \subseteq \mathcal{IR}$. From the definition of $\mathcal{C}_{b\text{-uni,uni}}^*$ in Eq. (5), we know that there is an epoch e_c before e, satisfying the adversary queries the challenge ciphertext in epoch e_c (i.e., $e_c \in \mathcal{C}$) and $\{e_c, \ldots, e\} \in \mathcal{T}_{b\text{-uni}}^*$, which implies $\{e_c, \ldots, e\} \in \mathcal{C}_{b\text{-uni}}^*$. From the assumption that $\mathcal{K}_{b\text{-uni}}^* \cap \mathcal{C}_{b\text{-uni,uni}}^* = \emptyset$, then we have $\{e_c, \ldots, e\} \notin \mathcal{K}_{b\text{-uni}}^*$. To meet the condition $\{e_c, \ldots, e\} \in \mathcal{T}_{b\text{-uni}}^*$,

all tokens in epochs in $\{e_c, \ldots, e\}$ can only be obtained by corrupting, that is $\{e_c, \ldots, e\} \in \mathcal{T} = \mathcal{T}_{no}^*$. We can upgrade the ciphertext from epoch e_c to epoch e_b even in the no-directional key update setting. Therefore, we have $e \in \mathcal{C}_{no,uni}^* \subseteq \mathcal{IR}$ and further $(c, e) \in \tilde{\mathcal{L}}_{no}^*$.

D.3   Proof of Theorem 2

We construct a reduction $\mathcal{B}_2$ who runs the $(b\text{-uni}, uni)\text{-notion}$ experiment and simulates all responses of the queries made by $(no, uni)\text{-notion}$ adversary $\mathcal{A}$. The reduction $\mathcal{B}_2$ works by sending all the queries of $\mathcal{A}$ to its own challenger and forwarding its received responses to $\mathcal{A}$. In the end, $\mathcal{B}_2$ sends the guessing result from $\mathcal{A}$ to its own challenger. The challenger will check if the reduction wins. The reduction also forwards the experiment result to $\mathcal{A}$. If the trivial win conditions were triggered, the reduction will be regarded as losing the game.

From Lemmas 8, 9, 10, 11 and Remark 3, we obtain the trivial win conditions in the $(b\text{-uni}, uni)\text{-notion}$ and $(no, uni)\text{-notion}$ games are equivalent. If there is a trivial win condition that is triggered by $\mathcal{A}$, then the same trivial win condition will be triggered by $\mathcal{B}$, and vice versa. Therefore, the reduction perfectly simulates the $(no, uni)\text{-notion}$ game to $\mathcal{A}$. Then, we have $\mathsf{Adv}_{UE,\mathcal{A}}^{(b\text{-uni},uni)\text{-notion}}(1^\lambda) = \mathsf{Adv}_{UE,\mathcal{B}_2}^{(no,uni)\text{-notion}}(1^\lambda)$.

D.4   Proof of Theorem 4

We construct a reduction $\mathcal{B}_4$ which runs the $(kk', cc')\text{-notion}$ game and simulates all responses to the queries made by the $(b\text{-uni}, cc)\text{-notion}$ adversary $\mathcal{A}$. If there is a trivial win condition that is triggered by $\mathcal{A}$, the same trivial win condition will be triggered by $\mathcal{B}$, and vice versa, which follows from Lemmas 12, 14 and 16. Thus, the reduction perfectly simulates the game to A, and the advantages are equal.

References

1. Barak, B., et al.: On the (im)possibility of obfuscating programs. J. ACM **59**(2), 1–48 (2012)
2. Boneh, D., Eskandarian, S., Kim, S., Shih, M.: Improving speed and security in updatable encryption schemes. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020. LNCS, vol. 12493, pp. 559–589. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64840-4_19
3. Boneh, D., Lewi, K., Montgomery, H., Raghunathan, A.: Key homomorphic PRFs and their applications. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8042, pp. 410–428. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40041-4_23
4. Boyd, C., Davies, G.T., Gjøsteen, K., Jiang, Y.: Fast and secure updatable encryption. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020. LNCS, vol. 12170, pp. 464–493. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-56784-2_16

5. Chen, L., Li, Y., Tang, Q.: CCA updatable encryption against malicious re-encryption attacks. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020. LNCS, vol. 12493, pp. 590–620. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64840-4_20

6. Everspaugh, A., Paterson, K., Ristenpart, T., Scott, S.: Key rotation for authenticated encryption. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10403, pp. 98–129. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63697-9_4

7. Galteland, Y.J., Pan, J.: Backward-leak uni-directional updatable encryption from public key encryption. Cryptology ePrint Archive, Paper 2022/324 (2022). https://eprint.iacr.org/2022/324

8. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions. J. ACM **33**(4), 792–807 (1986)

9. Günther, F., Hale, B., Jager, T., Lauer, S.: 0-RTT key exchange with full forward secrecy. In: Coron, J.-S., Nielsen, J.B. (eds.) EUROCRYPT 2017. LNCS, vol. 10212, pp. 519–548. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56617-7_18

10. Jiang, Y.: The direction of updatable encryption does not matter much. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020. LNCS, vol. 12493, pp. 529–558. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64840-4_18

11. Klooß, M., Lehmann, A., Rupp, A.: (R)CCA secure updatable encryption with integrity protection. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019. LNCS, vol. 11476, pp. 68–99. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17653-2_3

12. Lehmann, A., Tackmann, B.: Updatable encryption with post-compromise security. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018. LNCS, vol. 10822, pp. 685–716. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78372-7_22

13. Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: Ortiz, H. (ed.) STOC 1990, pp. 427–437. ACM, New York (1990)

14. Nishimaki, R.: The direction of updatable encryption does matter. In: Hanaoka, G., Shikata, J., Watanabe, Y. (eds.) PKC 2022. LNCS, vol. 13178, pp. 194–224. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-97131-1_7

15. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM **56**(6), 34:1–34:40 (2009)

16. Slamanig, D., Striecks, C.: Puncture 'em all: updatable encryption with no-directional key updates and expiring ciphertexts. Cryptology ePrint Archive, Paper 2021/268 (2021). https://eprint.iacr.org/2021/268