LETTER

On the Security of Keyed-Homomorphic PKE: Preventing Key Recovery Attacks and Ciphertext Validity Attacks*

Keita EMURA[†a)], ***Member***

SUMMARY In this short note, we formally show that Keyed-Homomorphic Public Key Encryption (KH-PKE) is secure against key recovery attacks and ciphertext validity attacks that have been introduced as chosen-ciphertext attacks for homomorphic encryption.

key words: *Keyed-Homomorphic PKE, CCA security, key recovery attacks, ciphertext validity attacks*

1. Introduction

Homomorphic encryption allows us to operate encrypted data, and this attractive functionality has been applied to construct several secure protocols, especially after the seminal work by Gentry [1]. However, the fact that homomorphic encryption schemes are vulnerable against (adaptive) chosen-ciphertext attacks (CCA)** is somewhat overlooked.

1.1 CCA1 Attacks against Homomorphic Encryption

Theoretically, an adversary sends a homomorphically evaluated challenge ciphertext to the decryption oracle, and can immediately break the security. One may think that this is just a theoretical result and there is no practical impact. Even though Bleichenbacher's CCA attack [2] has been widely recognized, it is also widely recognized that a weaker security level is acceptable in return for obtaining a homomorphic property. However, several CCA attacks against concrete homomorphic encryption schemes have been also shown. We introduce key recovery attacks (KRA) as follows.

Key Recovery Attacks: An adversary recovers secret keys via the access of the decryption oracle. These attacks work well regardless of whether or not a ciphertext input to the decryption oracle is the challenge one, and allows to recover secret keys. Currently, many attacks have been proposed [3]–[8]. We introduce these attacks in Sect. 1.4.

1.2 Is CCA1 Security Sufficient?

Canetti et al. [9] have proposed three IND-CCA1 secure fully homomorphic encryption schemes. Because decryption queries can be simulated, it might be sufficient to prevent key recovery attacks. However, CCA1 security seems to be insufficient owing to ciphertext validity attacks (CVA) as follows.

Ciphertext Validity Attacks: Although KRA is run via the access of the decryption oracle, CVA is run via the access of a ciphertext validity oracle (both before and after the challenge phase), where the oracle takes ciphertext as an input and determines whether it would output $\perp$ or not on decryption. We introduce the Loftus et al. attack [3] in Sect. 1.4.

1.3 Our Contribution

In this short note, we formally show that Keyed-Homomorphic Public Key Encryption (KH-PKE) [10]–[14], which is a CCA2 secure homomorphic encryption by introducing a designated evaluation, prevents key recovery attacks and ciphertext validity attacks. Since several homomorphic encryption schemes are vulnerable against these attacks as mentioned before, our result suggests that KH-PKE is an attractive option when the designated evaluation is allowable.

1.4 Related Work

Loftus et al. [3] have shown that the Gentry scheme (of the Gentry-Halevi variant [15]) is not CCA1 secure where a secret key $z \in [0, d)$ is recovered by $O(\log d)$ decryption queries. Zhang, Plantard, and Susilo [4] have shown a CCA1 attack against the Dijk-Gentry-Halevi-Vaikuntanathan scheme [16] that recovers a secret key by $O(\lambda^2)$ decryption queries (where λ is a security parameter). Chenal and Tang [5] have shown several key recovery attacks such as the Brakerski-Vaikuntanathan scheme [17] with nN decryption queries, where a secret key is an element of $\mathbb{Z}_q^n$ and $N = \lfloor \log_2(q-1) \rfloor + 1$, the other Brakerski-Vaikuntanathan scheme [18] with $\lceil \lfloor \log_2(q-1) \rfloor + 1/\lfloor \log_2(t-1) \rfloor + 1 \rceil$ decryption queries where $t = \mathsf{poly}(\lambda) \in \mathbb{Z}_q^*$, the Gentry-Sahai-Waters scheme [19] that each decryption query recovers 1 bit of each coefficient t_i of the secret vector $\vec{t} \in \mathbb{Z}_q^n$. They have also shown that these attacks work against the Brakerski-Gentry-Vaikuntanathan scheme [20]. The Dijk-Gentry-Halevi-Vaikuntanathan scheme [16] with $O(\eta)$ decryption queries where η is the bit-length of the secret key

Manuscript received April 10, 2020.
Manuscript revised May 21, 2020.
Manuscript publicized July 8, 2020.

[†] The author is with the National Institute of Communications and Information Technology (NICT), Koganei-shi, 184-8795 Japan.

[*] This work was supported by JST CREST Grant Number JPMJCR19F6, Japan.

a) E-mail: k-emura@nict.go.jp

DOI: 10.1587/transfun.2020EAL2039

[**] Throughout this study, we use CCA2 or CCA as adaptive chosen-ciphertext attacks and CCA1 as non-adaptive chosen-ciphertext (i.e., lunchtime) attacks.

(which improves the abovementioned Zhang et al.'s attack). Dahab, Galbraith, and Morais [6] have shown a key recovery attack against a NTRU-based scheme proposed by Bos et al. [21] with $d\lceil\log_2(B)\rceil$, where the scheme is constructed on a ring $\mathbb{Z}_q[x]/(x^d+1)$, d is a power of 2, B is a bound on the coefficient size of error distribution such that $B^2 < q/(36t^2)$, and t specifies a plaintext space R/tR with $R = \mathbb{Z}[x]/(x^d+1)$. Chenal and Tang [7] have shown a key recovery attack against an NTRU-based scheme proposed by Lopez-Alt et al. [22] with $\lfloor\log_2(B)\rfloor + n$ decryption queries, and improved the Dahab-Galbraith-Morais attack. Peng [8] showed a key recovery attack with single decryption query against the Brakerski-Fan-Vercauteren scheme (a Ring-LWE variant of the Brakerski scheme [23] proposed by Fan and Vercauteren [24]) which is employed in Microsoft SEAL.

Loftus et al. [3] have shown that the Smart-Vercauteren scheme [25] (with a modification by adding a ciphertext-checking procedure) is IND-CCA1 secure (under a lattice-based knowledge assumption), but is not IND-CVA secure. This is a CCA2-like attack where an adversary obtains the challenge ciphertext C^*, adds some values to C^* via homomorphic operations, and sends the ciphertext to the ciphertext validity oracle. They have shown that the decryption result of C^* is recovered by $O(N\log_2 T)$ ciphertext validity queries where N is the degree of a polynomial and T defines the size of the circuit. They insisted that "*Such an oracle can often be obtained in the real world by the attacker observing the behavior of a party who is fed ciphertexts of the attacker's choosing.*". Li, Galbraith, and Ma [26] also insisted that "*If a user is storing an encrypted database in the cloud and making queries to it, then an attacker could send ciphertexts of its choosing in response. If these ciphertexts are invalid, then the user might re-send the same query until a valid ciphertext is received in response. Such a situation precisely gives a CVA oracle.*".

2. Keyed-Homomorphic Public Key Encryption

Emura et al. [10], [11] have proposed a KH-PKE notion. In addition to a public and decryption key pair (pk, sk_d), a homomorphic operation key sk_h is defined and the evaluation algorithm requires to take sk_h as input while anyone can evaluate ciphertext freely in usual homomorphic encryption schemes. This designated evaluation allows us to define CCA security for outsiders who do not have sk_h.

Emura et al. have shown two instantiations of their generic construction. The first one is a multiplicative KH-PKE scheme, which is secure under the decisional Diffie-Hellman (DDH) assumption, and the second one is an additive KH-PKE scheme, which is secure under the decisional composite residuosity (DCR) assumption. These schemes are pairing-free. Later, Libert, Peters, Joye, and Yung (LPYJ) [12] have proposed a multiplicative KH-PKE scheme supporting threshold decryption and publicly verifiability. Jutla and Roy (JR) [13] have also proposed a publicly verifiable KH-PKE scheme with a shorter ciphertext size. The LPYJ and JR KH-PKE schemes are pairing-based. Though these KH-PKE schemes support either additive or multiplicative homomorphic operation, Lai et al. [14] have proposed Keyed-Fully Homomorphic Encryption (keyed-FHE) whose security relies on the learning with errors (LWE) assumption and indistinguishability obfuscation ($i\mathcal{O}$) [27]. Thus, one definite future work is to construct a keyed-FHE scheme without $i\mathcal{O}$. We summarize these schemes in Table 1.

Table 1 KH-PKE schemes.

	Homomorphism	Assumptions
Emura et al. [10], [11]	Multiplicative	DDH
Emura et al. [10], [11]	Additive	DCR
LPJY [12]	Multiplicative	DLIN
JR [13]	Multiplicative	SXDH
LDMSW [14]	Full	LWE & $i\mathcal{O}$

The syntax of KH-PKE is given as follows.

Definition 1 (Syntax of KH-PKE [10], [11]): Let $\mathcal{M}$ be a message space and $\odot$ be a binary operation over $\mathcal{M}$. A KH-PKE scheme $\mathcal{KH\text{-}PKE}$ = (KeyGen, Enc, Dec, Eval) for homomorphic operation $\odot$ consists of the following four algorithms:

KeyGen**:** The key generation algorithm takes a security parameter $\lambda \in \mathbb{N}$ as input, and outputs a public key pk, a decryption key sk_d, and a homomorphic operation key sk_h.

Enc**:** The encryption algorithm takes pk, and a message $M \in \mathcal{M}$ as input, and outputs a ciphertext C.

Dec**:** The decryption algorithm takes sk_d and C as input, and outputs M or $\perp$.

Eval**:** The evaluation algorithm takes sk_h, two ciphertexts C_1 and C_2 as input, and outputs a ciphertext C or $\perp$.

Next, we provide the definition of correctness. For a public key pk generated by the KeyGen algorithm, let $C_{pk,M}$ be the set of all ciphertexts of $M \in \mathcal{M}$ under pk.

Definition 2 (Correctness [10], [11]): We say that the KH-PKE scheme for homomorphic operation $\odot$ is *correct* if for all $(pk, sk_d, sk_h) \leftarrow \mathsf{KeyGen}(1^\lambda)$, the following two conditions hold: (1) For all $M \in \mathcal{M}$, and all $C \in C_{pk,M}$, $\mathsf{Dec}(sk_d, C) = M$ holds. (2) For all $M_1, M_2 \in \mathcal{M}$, all $C_1 \in C_{pk,M_1}$, and all $C_2 \in C_{pk,M_2}$, $\mathsf{Eval}(sk_h, C_1, C_2) \in C_{pk,M_1\odot M_2}$ holds.

Next, we provide the definition of indistinguishability under adaptive chosen ciphertext attacks (IND-KH-CCA). We simply denote IND-KH-CCA as KH-CCA.

Definition 3 (KH-CCA): We say that the KH-PKE scheme is KH-CCA secure if for any probabilistic polynomial-time (PPT) adversary $\mathcal{A}$, the advantage

$$\begin{aligned} Adv^{\mathsf{KH\text{-}CCA}}_{\mathsf{KH\text{-}PKE},\mathcal{A}}(\lambda) = |\Pr[&(pk, sk_d, sk_h) \leftarrow \mathsf{KeyGen}(1^\lambda); \\ &(M_0^*, M_1^*, State) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{find}, pk);\ \beta \overset{\$}{\leftarrow} \{0,1\}; \\ &C^* \leftarrow \mathsf{Enc}(pk, M_\beta^*);\ \beta' \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{guess}, State, C^*); \\ &\beta = \beta'] - \frac{1}{2}| \end{aligned}$$

is negligible in λ, where O consists of oracles RevHK, $\mathsf{Eval}(sk_h, \cdot, \cdot)$, and $\mathsf{Dec}(sk_d, \cdot)$ defined as follows. Let $\mathcal{D}$ be a list which is initialized as $\emptyset$, and is set as $\mathcal{D} = \{C^*\}$ right after the challenge stage.

- RevHK: Upon a request, the homomorphic key reveal oracle responds with sk_h. This oracle is available only once.
- $\mathsf{Eval}(sk_h, \cdot, \cdot)$: If RevHK has already been queried before, then the evaluation oracle is not available. Otherwise, the oracle responds to a query (C_1, C_2) with the result of $C \leftarrow \mathsf{Eval}(sk_h, C_1, C_2)$. In addition, if $C \neq \perp$ and either $C_1 \in \mathcal{D}$ or $C_2 \in \mathcal{D}$, then the oracle updates the list by $\mathcal{D} \leftarrow \mathcal{D} \cup \{C\}$.
- $\mathsf{Dec}(sk_d, \cdot)$: The decryption oracle is not available if $\mathcal{A}$ has queried to RevHK *and* $\mathcal{A}$ has obtained the challenge ciphertext C^*. Otherwise, the oracle responds to a query C with the result of $\mathsf{Dec}(sk_d, C)$ if $C \notin \mathcal{D}$ or returns $\perp$ otherwise.

It is particularly worth noting that any ciphertext including the challenge one is allowed to be the inputs of the evaluation oracle. As a restriction to avoid trivial attacks, the challenge ciphertext and challenge-related ciphertexts (which are listed in $\mathcal{D}$) are not allowed to be input into the decryption oracle.

Is Designated Evaluation Setting Sufficient? In KH-PKE, the evaluation algorithm requires sk_h. This designated evaluation setting is acceptable in the following case: (1) a client sets up a public and decryption key pair, encrypts data, and sends the ciphertext to a server, (2) the server runs the evaluation algorithm for encrypted data to perform homomorphic operations to them, and returns the evaluation result (a ciphertext) to the client, and (3) the client decrypts the ciphertext and obtains the result. As an example of this framework, Shimizu et al. [28] have proposed a privacy-preserving search mechanism for chemical compound databases using the additive homomorphic encryption.

3. On the Security of KH-PKE

Although KH-CCA security formally states the security of KH-PKE, in this section we formally show that KH-CCA can prevent key recovery attacks and ciphertext validity attacks. We assume that an adversary $\mathcal{A}$ is an outsider who does not call the RevHK oracle.

3.1 KH-PKE is Secure against Key Recovery Attacks

Here, we formalize key recovery attacks in the KH-PKE context. We assume that $\mathcal{A}$ recovers the actual sk_d for the sake of simplicity. However, we can easily extend it such that $\mathcal{A}$ recovers an equivalent key $sk'_d \neq sk_d$ where for all $M \in \mathcal{M}$, and all $C \in \mathcal{C}_{pk,M}$, $\mathsf{Dec}(sk'_d, C) = M$ holds.

Definition 4 (KRA Security): We say that the KH-PKE scheme is secure against key recovery attacks if for any PPT adversary $\mathcal{A}$, the advantage

$$Adv^{\mathsf{KRA}}_{\mathsf{KH\text{-}PKE},\mathcal{A}}(\lambda) = \Pr[(pk, sk_d, sk_h) \leftarrow \mathsf{KeyGen}(1^\lambda);\ sk_d \leftarrow \mathcal{A}^{\mathsf{Dec}(sk_d,\cdot)}(pk)]$$

is negligible in λ, where $\mathsf{Dec}(sk_d, \cdot)$ is the decryption oracle which responds to a query C with the result of $\mathsf{Dec}(sk_d, C)$.

Next, we show that if the KH-PKE scheme is KH-CCA secure, then it is secure against key recovery attacks.[†] This is somewhat trivial since the decryption oracle is available in the definition of KH-CCA, and all decryption queries are simulatable. Let $\mathcal{A}$ be an adversary that issues decryption queries and recovers the secret key sk_d. We show there exists an algorithm $\mathcal{B}$ that breaks the KH-CCA security of a KH-PKE scheme by interacting with $\mathcal{A}$. First, $\mathcal{B}$ receives pk from the KH-CCA challenger of the KH-PKE scheme. $\mathcal{B}$ forwards it to $\mathcal{A}$. When $\mathcal{A}$ sends a decryption query, then $\mathcal{B}$ forwards it to the challenger, obtains the decryption result, and returns it to $\mathcal{A}$. Since no target ciphertext exists in key recovery attacks, $\mathcal{A}$ outputs sk_d at some point. Then, $\mathcal{B}$ chooses (M_0^*, M_1^*) and sends it to the challenger, and the challenger returns the challenge ciphertext C^*.[††] By decrypting C^*, $\mathcal{B}$ breaks the KH-CCA security. This shows that if the KH-PKE scheme is KH-CCA secure, then it is KRA secure.

3.2 KH-PKE is Secure against Ciphertext Validity Attacks

Here, we formalize ciphertext validity attacks in the KH-PKE context. Since this is a CCA2-like attack, we additionally consider the evaluation oracle that allows the adversary to check the validity of challenge-related ciphertexts. Although here we give an IND-type definition, Loftus et al. [3] gave an onewayness-type definition: the challenge ciphertext C^* is associated with a hidden message M^* and $\mathcal{A}$ recovers M^*. Our IND-type definition implies the onewayness-type definition.

Definition 5 (IND-CVA): We say that the KH-PKE scheme is IND-CVA secure if for any PPT adversary $\mathcal{A}$, the advantage

$$\begin{aligned} Adv^{\mathsf{IND\text{-}CVA}}_{\mathsf{KH\text{-}PKE},\mathcal{A}}(\lambda) = |\Pr[&(pk, sk_d, sk_h) \leftarrow \mathsf{KeyGen}(1^\lambda);\\ &(M_0^*, M_1^*, State) \leftarrow \mathcal{A}^{\mathsf{Validity}(sk_d,\cdot),\mathsf{Eval}(sk_h,\cdot,\cdot)}(\mathsf{find}, pk);\\ &\beta \xleftarrow{\$} \{0,1\};\ C^* \leftarrow \mathsf{Enc}(pk, M_\beta^*);\\ &\beta' \leftarrow \mathcal{A}^{\mathsf{Validity}(sk_d,\cdot),\mathsf{Eval}(sk_h,\cdot,\cdot)}(\mathsf{guess}, State, C^*);\\ &\beta = \beta'] - \frac{1}{2}| \end{aligned}$$

is negligible in λ, where the ciphertext validity oracle $\mathsf{Validity}(sk_d, \cdot)$ is defined as follows.

- $\mathsf{Validity}(sk_d, \cdot)$: The oracle takes as input C and returns 1 if the result of $\mathsf{Dec}(sk_d, C)$ is not $\perp$ or returns 0

[†] Our reduction still works in the equivalent key case. Moreover, our reduction works even if $\mathcal{A}$ is allowed to access the evaluation oracle.

[††] Clearly, our reduction works even if $\mathcal{A}$ chooses (M_0^*, M_1^*).

otherwise.

Basically, the ciphertext validity oracle is directly simulated by the decryption oracle, i.e., if the decryption oracle returns a non-$\perp$ value, then the reduction returns 1, and otherwise, if the decryption oracle returns $\perp$, then the reduction returns 0. Because Das et al. [29] have shown that CVA is weaker than CCA2, one may think that it is also trivial to show that a KH-CCA secure KH-PKE scheme is also secure against ciphertext validity attacks. However, we need to additionally simulate the ciphertext validity oracle even when challenge-related ciphertexts containing $\mathcal{D}$ are queried (remember that in the Loftus et al. attack the challenge ciphertext is modified via homomorphic operations, and the modified ciphertext is sent to the ciphertext validity oracle). Of note, a KH-CCA adversary (the reduction in this context) is not allowed to send a ciphertext $C \in \mathcal{D}$ to the decryption oracle to avoid a trivial attack. However, such ciphertexts are generated via the evaluation oracle, and thus the reduction simply returns 1 if $C \in \mathcal{D}$ is queried to the ciphertext validity oracle. We emphasize that if a challenge-related ciphertext $C \notin \mathcal{D}$ is queried (it shows that homomorphic operations were done without using sk_h), then it immediately breaks the KH-CCA security. Thus, here, we are interested in the case $C \in \mathcal{D}$ only for handling challenge-related queries.

Next, we show that if the KH-PKE scheme is KH-CCA secure, then it is secure against ciphertext validity attacks. Let $\mathcal{A}$ be an adversary that issues ciphertext validity queries. $\mathcal{A}$ is additionally allowed to access the evaluation oracle. We show there exists an algorithm $\mathcal{B}$ that breaks the KH-CCA security of a KH-PKE scheme by interacting with $\mathcal{A}$. First, $\mathcal{B}$ receives pk from the KH-CCA challenger of the KH-PKE scheme. $\mathcal{B}$ forwards it to $\mathcal{A}$. If $\mathcal{A}$ sends a ciphertext validity query C, then $\mathcal{B}$ forwards it to the challenger as a decryption query, obtains the decryption result M, and returns 1 if $M \neq \perp$ and 0 if $M = \perp$. If $\mathcal{A}$ sends an evaluation query (C_1, C_2) to $\mathcal{B}$, then $\mathcal{B}$ forwards it to the challenger as an evaluation query, and returns the evaluation resut C to $\mathcal{A}$. In the challenge phase, $\mathcal{A}$ chooses (M_0^*, M_1^*) and sends it to $\mathcal{B}$. $\mathcal{B}$ forwards it to the challenger, obtains the challenge ciphertext C^*, and returns it to $\mathcal{A}$. Later, $\mathcal{B}$ also manages a set $\mathcal{D}'$ which is initialized as $\{C^*\}$. If $\mathcal{A}$ sends an evaluation query (C_1, C_2) to $\mathcal{B}$, then $\mathcal{B}$ forwards it to the challenger as an evaluation query, and returns the evaluation result C to $\mathcal{A}$. If $C \neq \perp$ and either $C_1 \in \mathcal{D}'$ or $C_2 \in \mathcal{D}'$, then $\mathcal{B}$ updates the list by $\mathcal{D}' \leftarrow \mathcal{D} \cup \{C\}$. If $\mathcal{A}$ sends a ciphertext validity query $C \notin \mathcal{D}'$, then $\mathcal{B}$ forwards it to the challenger as a decryption query, obtains the decryption result M, and returns 1 if $M \neq \perp$ and 0 if $M = \perp$. If $\mathcal{A}$ sends a ciphertext validity query $C \in \mathcal{D}'$, then $\mathcal{B}$ returns 1. $\mathcal{A}$ outputs β' at some point. Then, $\mathcal{B}$ outputs β', and breaks the KH-CCA security. This shows that if the KH-PKE scheme is KH-CCA secure, then it is IND-CVA secure.[†]

4. Discussion

Differences Other Post-CCA1 Security: Loftus et al. [3] have introduced the notion of CCA-embeddable homomorphic encryption. From a ciphertext of a CCA secure PKE scheme (e.g., the Cramer-Shoup PKE scheme [30]), anyone can extract a ciphertext of a CPA homomorphic PKE scheme (e.g., the ElGamal PKE scheme). Unfortunately, CCA security is missing after homomorphic operations. Li, Galbraith, and Ma [26] have modified the Gentry-Sahai-Waters scheme [19] where the decryption algorithm generates a fresh random one-time secret key for each decryption. They insisted that the scheme is secure against ciphertext validity attacks because no valid ciphertext notion is introduced (i.e, the decryption algorithm does not output $\perp$). Owing to the Chenal-Tang key recovery attack against the Gentry-Sahai-Waters scheme [5], some bits of the one-time private key are recovered by decryption queries, however, this does not allow the adversary to compute a valid secret key owing to the freshness. Although this attempt is attractive, it is not proved that the modified Gentry-Sahai-Waters scheme is IND-CCA1 secure. Desmedt et al. [31] have proposed controlled homomorphic encryption where a token is required for evaluation. This also introduces the designated evaluation setting but does not consider any CCA security. Joo and Yun [32] proposed homomorphic authenticated encryption and defined its CCA security. Unlike KH-PKE, it is symmetric key encryption.

Is Double Encryption Sufficient? Emura et al. have shown that a simple double encryption produces the designated evaluation where a plaintext is encrypted by a CCA1 homomorphic PKE scheme, and the ciphertext is again encrypted by a CCA2 PKE scheme. Then the decryption key of the CCA2 PKE scheme is regarded as sk_h. It may be sufficient to prevent key recovery attacks, but is insufficient to prevent ciphertext validity attacks. Let the inner PKE be CCA1 secure but not CVA secure (as the modified Smart-Vercauteren scheme [3]). After obtaining the challenge ciphertext, if a CVA adversary calls the ciphertext validity oracle, then the scheme is broken. If a CVA secure PKE scheme, or rather CCA1.5 secure PKE scheme[††], is employed as the inner PKE, then it may be sufficient to protect the scheme against outsider adversaries. However, KH-CCA security considers the following case, which is not captured by double encryption: an adversary issues decryption queries even after the challenge phase and later the adversary obtains sk_h via the RevHK oracle. Although there is a room for argument on whether or not this case affects security in practice, nevertheless, we insist that stronger security should be considered as much as possible as long as the homomorphic functionality is provided.

[†] Clearly, our reduction works even if $\mathcal{A}$ is allowed to access the decryption oracle.

[††] Das et al. [29] defined CCA1.5 where the decryption oracle is available before the challenge phase and the ciphertext validity oracle is available even after the challenge phase. They showed that CCA1.5 is stronger than CCA1 but is weaker than (replayable) CCA2.

References

[1] C. Gentry, "Fully homomorphic encryption using ideal lattices," STOC, pp.169–178, 2009.

[2] D. Bleichenbacher, "Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1," CRYPTO, pp.1–12, 1998.

[3] J. Loftus, A. May, N.P. Smart, and F. Vercauteren, "On CCA-secure somewhat homomorphic encryption," Selected Areas in Cryptography, pp.55–72, 2011.

[4] Z. Zhang, T. Plantard, and W. Susilo, "On the CCA-1 security of somewhat homomorphic encryption over the integers," ISPEC, pp.353–368, 2012.

[5] M. Chenal and Q. Tang, "On key recovery attacks against existing somewhat homomorphic encryption schemes," LATINCRYPT, pp.239–258, 2014.

[6] R. Dahab, S.D. Galbraith, and E. Morais, "Adaptive key recovery attacks on NTRU-based somewhat homomorphic encryption schemes," ICITS, pp.283–296, 2015.

[7] M. Chenal and Q. Tang, "Key recovery attacks against NTRU-based somewhat homomorphic encryption schemes," ISC, pp.397–418, 2015.

[8] Z. Peng, "Danger of using fully homomorphic encryption: A look at microsoft SEAL," CoRR, vol.abs/1906.07127, 2019.

[9] R. Canetti, S. Raghuraman, S. Richelson, and V. Vaikuntanathan, "Chosen-ciphertext secure fully homomorphic encryption," Public-Key Cryptography, pp.213–240, 2017.

[10] K. Emura, G. Hanaoka, G. Ohtake, T. Matsuda, and S. Yamada, "Chosen ciphertext secure keyed-homomorphic public-key encryption," Public-Key Cryptography, pp.32–50, 2013.

[11] K. Emura, G. Hanaoka, K. Nuida, G. Ohtake, T. Matsuda, and S. Yamada, "Chosen ciphertext secure keyed-homomorphic public-key cryptosystems," Des. Codes Cryptogr., vol.86, no.8, pp.1623–1683, 2018.

[12] B. Libert, T. Peters, M. Joye, and M. Yung, "Non-malleability from malleability: Simulation-sound quasi-adaptive NIZK proofs and CCA2-secure encryption from homomorphic signatures," EUROCRYPT, pp.514–532, 2014.

[13] C.S. Jutla and A. Roy, "Dual-system simulation-soundness with applications to UC-PAKE and more," ASIACRYPT, pp.630–655, 2015.

[14] J. Lai, R.H. Deng, C. Ma, K. Sakurai, and J. Weng, "CCA-secure keyed-fully homomorphic encryption," Public-Key Cryptography, pp.70–98, 2016.

[15] C. Gentry and S. Halevi, "Implementing Gentry's fully-homomorphic encryption scheme," EUROCRYPT, pp.129–148, 2011.

[16] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," EUROCRYPT, pp.24–43, 2010.

[17] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," FOCS, pp.97–106, 2011.

[18] Z. Brakerski and V. Vaikuntanathan, "Fully homomorphic encryption from ring-lwe and security for key dependent messages," CRYPTO, pp.505–524, 2011.

[19] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based," CRYPTO, pp.75–92, 2013.

[20] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," ITCS, pp.309–325, 2012.

[21] J.W. Bos, K.E. Lauter, J. Loftus, and M. Naehrig, "Improved security for a ring-based fully homomorphic encryption scheme," IMACC, pp.45–64, 2013.

[22] A. López-Alt, E. Tromer, and V. Vaikuntanathan, "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption," STOC, pp.1219–1234, 2012.

[23] Z. Brakerski, "Fully homomorphic encryption without modulus switching from classical gapsvp," CRYPTO, pp.868–886, 2012.

[24] J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption," IACR Cryptology ePrint Archive, vol.2012, p.144, 2012.

[25] N.P. Smart and F. Vercauteren, "Fully homomorphic encryption with relatively small key and ciphertext sizes," Public Key Cryptography, pp.420–443, 2010.

[26] Z. Li, S.D. Galbraith, and C. Ma, "Preventing adaptive key recovery attacks on the GSW levelled homomorphic encryption scheme," Provable Security, pp.373–383, 2016.

[27] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S.P. Vadhan, and K. Yang, "On the (im)possibility of obfuscating programs," J. ACM, vol.59, no.2, p.6, 2012.

[28] K. Shimizu, K. Nuida, H. Arai, S. Mitsunari, N. Attrapadung, M. Hamada, K. Tsuda, T. Hirokawa, J. Sakuma, G. Hanaoka, and K. Asai, "Privacy-preserving search for chemical compound databases," BMC Bioinform., vol.16, no.S6, 2015.

[29] A. Das, S. Dutta, and A. Adhikari, "Indistinguishability against chosen ciphertext verification attack revisited: The complete picture," Provable Security, pp.104–120, 2013.

[30] R. Cramer and V. Shoup, "A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack," CRYPTO, pp.13–25, 1998.

[31] Y. Desmedt, V. Iovino, G. Persiano, and I. Visconti, "Controlled homomorphic encryption: Definition and construction," Workshop on Encrypted Computing and Applied Homomorphic Cryptography, pp.107–129, 2017.

[32] C. Joo and A. Yun, "Homomorphic authenticated encryption secure against chosen-ciphertext attack," ASIACRYPT, pp.173–192, 2014.