Homomorphic Authenticated Encryption Secure Against Chosen-Ciphertext Attack

Chihong Joo and Aaram Yun

Ulsan National Institute of Science and Technology (UNIST)
Republic of Korea
{chihongjoo,aaramyun}@unist.ac.kr

Abstract. We study *homomorphic authenticated encryption*, where privacy and authenticity of data are protected simultaneously. We define homomorphic versions of various security notions for privacy and authenticity, and investigate relations between them. In particular, we show that it is possible to give a natural definition of IND-CCA for homomorphic authenticated encryption, unlike the case of homomorphic encryption. Also, we construct a simple homomorphic authenticated encryption scheme supporting arithmetic circuits, which is chosen-ciphertext secure both for privacy and authenticity. Our scheme is based on the error-free approximate GCD assumption.

Keywords: homomorphic authenticated encryption, homomorphic MAC, homomorphic encryption

1 Introduction

Homomorphic cryptography allows processing of cryptographically protected data. For example, homomorphic encryption lets a third party which does not have the secret key to evaluate functions implicitly using only ciphertexts so that the computed ciphertext decrypts to the correct function value. Similarly, homomorphic signature allows a third party who is not the signer to derive a signature to the output of a function, given signatures of the inputs. This possibility for secure delegation of computation could potentially be used for many applications including cloud computing, and so it makes homomorphic cryptography a very interesting area, which was recently attracting many focused research activities, especially since Gentry's first construction [20] of fully homomorphic encryption (FHE) in 2009. While existing FHE schemes are still many orders slower than ordinary encryption schemes to be truly practical, many progresses are already being made in improving the efficiency of FHE [17, 27, 9, 10, 21, 15, 16, 6, 8, 22, 23, 13, 7, 14]. Eventually, a truly practical FHE could be used to build secure cloud computing services where even the cloud provider cannot violate the privacy of the data stored and processed by the cloud.

But, if such data is important enough to protect its privacy, in many scenarios the authenticity of the data would also be worth protecting simultaneously. Indeed, in symmetric-key cryptography, the authenticated encryption [26, 4, 18, 25] is exactly such a primitive protecting both. Therefore, we would like to study *homomorphic authenticated encryption* (henceforth abbreviated as HAE), which is a natural analogue of the authenticated encryption for homomorphic cryptography. A HAE is a symmetric-key primitive which allows public evaluation of functions using only corresponding ciphertexts.

Just as in the case of homomorphic encryption, one important goal in this area would be to design a *fully homomorphic* authenticated encryption. Since there are several known FHE constructions, we may construct a fully homomorphic authenticated encryption scheme by generic composition [4] with a fully homomorphic signature, or even a fully homomorphic MAC. But until very recently, the homomorphic signature scheme closest to being fully homomorphic was [5], where only low-degree polynomial functions are supported. A fully homomorphic MAC is proposed by Gennaro and Wichs [19], but it supports only a limited number of verification queries, so that the solution is in a sense incomplete. So far, the problem of constructing a fully homomorphic authenticated encryption is still not solved completely satisfyingly.[1]

Our contribution in this paper is twofold. First, we define various security notions for HAE and study relations among them. For privacy, we define homomorphic versions of IND-CPA and IND-CCA. While IND-CCA is not achievable for homomorphic encryption due to malleability, nevertheless we may define a version of IND-CCA for HAE. It is because that for HAE, encryption of a plaintext is done with respect to a 'label', and similarly decryption of a ciphertext is done with respect to a 'labeled program'. So, while the ciphertext is still malleable by function evaluation, a decryption query should essentially declare how the ciphertext was produced. This allows a homomorphic version of IND-CCA to be defined naturally.

For authenticity, we define UF-CPA, the homomorphic version of the unforgeability when the adversary has access to the encryption oracle. We also consider UF-CCA, where the adversary has both the encryption oracle and the decryption oracle. Moreover, we consider strong unforgeability flavors of authenticity, and define homomorphic versions accordingly: SUF-CPA and SUF-CCA. We investigate relationship between these notions, and, for example, show that SUF-CPA implies SUF-CCA. And, we show that IND-CPA and SUF-CPA imply IND-CCA. Together, this shows that a HAE scheme with IND-CPA and SUF-CPA security is in fact IND-CCA and SUF-CCA.

The second contribution is that we propose a HAE scheme supporting arithmetic circuits. This scheme is somewhat homomorphic and not fully homomorphic, but we show that our scheme is secure and satisfies both IND-CCA and SUF-CCA. Another appeal of our scheme is that it is a straightforward construction based on the error-free approximate GCD (EF-AGCD) assumption. EF-AGCD assumption was used before [17, 15, 16, 13, 14] in constructing fully

[1] Very recently, some constructions of leveled fully homomorphic signature schemes are proposed [28, 24], after the current paper has been submitted to Asiacrypt. So, at least the fully homomorphic authenticated encryption via generic composition would be possible now.

homomorphic encryption schemes supporting boolean circuits, but here we use it to construct a HAE scheme supporting arithmetic circuits on $\mathbb{Z}_Q$ for $Q \in \mathbb{Z}^+$.

2 Related Work

After this paper has been submitted to Asiacrypt, there were many progresses in the area of homomorphic signatures. In CRYPTO 2014, Catalano, Fiore, and Warinschi constructed homomorphic signature schemes for polynomial functions [12]. Compared with the scheme of Boneh and Freeman [5], their construction is in the standard model, and allows efficient verification.

Also, more relevantly, constructions of (leveled) fully homomorphic signature schemes are proposed by Wichs [28] and also by Gorbunov and Vaikuntanathan [24]. Therefore, the fully homomorphic authenticated encryption scheme via generic composition would be now possible, using these. However, currently known FHEs require large amount of ciphertext expansion, and that would become worse by generic composition. Designing more efficient fully homomorphic authenticated encryption would be an interesting problem.

Gennaro and Wichs [19] proposed the first construction of the fully homomorphic MAC. Their construction uses FHE, and exploits randomness in the encryption to hide data necessary for authentication. In fact, since their scheme encrypts plaintexts using FHE, it is already a fully homomorphic authenticated encryption. But, their construction essentially does not allow verification queries, so it satisfies only weaker security notions: IND-CPA and UF-CPA, according to our definition.

Catalano and Fiore [11] proposed two somewhat homomorphic MACs supporting arithmetic circuits on $\mathbb{Z}_p$ for prime modulus p. In their construction, a MAC for a message m is a polynomial $\sigma(X)$ such that its constant term $\sigma(0)$ is equal to the message m, and its value $\sigma(\alpha)$ on a secret random point α is equal to randomness determined by the label τ of the message m. While their construction is very simple and practical, it does not protect privacy of data, and it seems that this cannot be changed by simple modifications, for example by choosing a secret random value β as the value satisfying $\sigma(\beta) = m$. Also, the size of the modulus p is related to the security of the scheme, so it cannot be chosen arbitrarily.

Our scheme is not as efficient as the schemes of Catalano and Fiore, but certainly more efficient than the generically composed HAE of a FHE scheme and the Catalano-Fiore homomorphic MAC. And our scheme is also relatively straightforward. Moreover, in our construction, the security does not depend on the modulus Q so that it can be chosen arbitrarily depending on the application.

Our scheme can also be compared with a homomorphic encryption scheme called IDGHV presented in [13]. It supports encryption of a plaintext vector $(m_1, \ldots, m_\ell)$ where each m_i is an element in $\mathbb{Z}_{Q_i}$. Like our scheme, IDGHV also uses the Chinese remainder theorem, and indeed our construction can be seen as a special-case, symmetric-key variant of IDGHV where $\ell = 1$, and where encryption randomness is pseudorandomly generated from the label. We intentionally omitted encryption of multiple plaintexts for simplicity of exposition, but our construction can naturally be extended in this way.

Security notions of the authenticated encryption was studied before. Bellare and Namprempre [4] studied both privacy and authenticity of authenticated encryption schemes, and the authenticity notions are later studied further by Bellare, Goldreich and Mityagin [2]. Our UF-CPA and SUF-CPA can be considered as homomorphic versions of INT-PTXT-1 and INT-CTXT-1 of [2], respectively. Our UF-CCA and SUF-CCA are comparable to homomorphic versions of INT-PTXT-M and INT-CTXT-M, respectively, but in our (S)UF-CCA, the adversary has access to the decryption oracle, while in INT-PTXT-M and INT-CTXT-M, the adversary has access to the verification oracle.

3 Preliminary

3.1 Notations

For any $a \in \mathbb{R}$, the *nearest integer* $\lfloor a \rceil$ of a is defined as the unique integer in $\left[a - \frac{1}{2}, a + \frac{1}{2}\right)$. The ring $\mathbb{Z}_n$ of integers modulo n is represented as the set $\mathbb{Z} \cap (-\frac{n}{2}, \frac{n}{2}]$. That is, $x \bmod n = x - \lfloor x/n \rceil \cdot n$ for any $x \in \mathbb{Z}$. For example, we have $\mathbb{Z}_2 = \{0, 1\}$, $\mathbb{Z}_3 = \{-1, 0, 1\}$.

For any mutually prime $n, m \in \mathbb{Z}^+$, $\mathrm{CRT}_{(n,m)}$ is the isomorphism $\mathbb{Z}_n \times \mathbb{Z}_m \to \mathbb{Z}_{nm}$, such that for any $(a, b) \in \mathbb{Z}_n \times \mathbb{Z}_m$, we have

$$(\mathrm{CRT}_{(n,m)}(a, b) \bmod n, \mathrm{CRT}_{(n,m)}(a, b) \bmod m) = (a, b) \ .$$

In this paper, the security parameter is always denoted as λ, and the expression $f(\lambda) = \mathit{negl}(\lambda)$ means that $f(\lambda)$ is a negligible function, that is, for any $c > 0$, we have $|f(\lambda)| \le \lambda^{-c}$ for all $\lambda \in \mathbb{Z}^+$ large enough.

Also, lg means the logarithm to base 2. And $\Delta(D_1, D_2)$ denotes the statistical distance between two distributions D_1 and D_2.

A notation like $(\tau, \cdot) \in S$ is an abbreviation for $\exists x\, (\tau, x) \in S$. Naturally, $(\tau, \cdot) \notin S$ is its negation, $\forall x\, (\tau, x) \notin S$. This notation can also be generalized to n-tuples for $n > 2$, for example $(\tau, \cdot, \cdot) \in S$.

3.2 Security Assumptions

Here we define security assumptions we use in this paper. First, let us define some distributions. For $p, q_0, \rho \in \mathbb{Z}^+$, we define the distribution $\mathcal{D}(p, q_0, \rho)$ as

$$\mathcal{D}(p, q_0, \rho) := \{\text{choose } q \xleftarrow{\$} \mathbb{Z} \cap [0, q_0),\ r \xleftarrow{\$} \mathbb{Z} \cap (-2^\rho, 2^\rho) :\ \text{output } pq + r\} \ .$$

Clearly, we can efficiently sample from the above distribution. In this paper, when a distribution is given as an input to an algorithm, it means that a sampling oracle for the distribution is given.

Let PRIME be the set of all prime numbers, and ROUGH(x) the set of all 'x-rough integers', that is, integers having no prime factors less than x.

In the following, the parameters ρ, η, γ are polynomially bounded functions of λ, and we assume they can be efficiently computed, given λ.

Definition 1 (Error-Free Approximate GCD Assumption). *The (computational) (ρ, η, γ)-EF-AGCD assumption is that, for any PPT adversary A, we have*

$$\Pr\left[A(1^\lambda, y_0, \mathcal{D}(p, q_0, \rho)) = p\right] = negl(\lambda) \ ,$$

where $p \xleftarrow{\$} [2^{\eta-1}, 2^\eta) \cap \mathrm{PRIME}$, $q_0 \xleftarrow{\$} [0, 2^\gamma/p) \cap \mathrm{ROUGH}(2^{\lambda^2})$, and $y_0 = pq_0$.

The EF-AGCD assumption is suggested by Coron et al. [15] to prove the security of their variant of the DGHV scheme [17]. There is also a decisional version, suggested by Cheon at al. [13]:

Definition 2 (Decisional Error-Free Approximate GCD Assumption). *The decisional (ρ, η, γ)-EF-AGCD assumption is that, for any PPT distinguisher D, the following value is negligible:*

$$\Big| \Pr\big[D(1^\lambda, y_0, \mathcal{D}(p, q_0, \rho), z) = 1 \mid z \leftarrow \mathcal{D}(p, q_0, \rho)\big] \\ - \Pr\big[D(1^\lambda, y_0, \mathcal{D}(p, q_0, \rho), z) = 1 \mid z \xleftarrow{\$} \mathbb{Z}_{y_0}\big] \Big| \ ,$$

where $p \xleftarrow{\$} [2^{\eta-1}, 2^\eta) \cap \mathrm{PRIME}$, $q_0 \xleftarrow{\$} [0, 2^\gamma/p) \cap \mathrm{ROUGH}(2^{\lambda^2})$, and $y_0 = pq_0$.

Recently Coron et al. proved the equivalence of the EF-AGCD assumption and the decisional EF-AGCD assumption [14]. In Theorem 6, we show that our scheme is IND-CPA under the decisional EF-AGCD assumption. Hence, our scheme's security is in fact based on the (computational) EF-AGCD assumption, due to the equivalence.

4 Homomorphic Authenticated Encryption

Here we define the homomorphic authenticated encryption and its security. In the following, $\mathcal{M}$, $\mathcal{C}$, $\mathcal{L}$, $\mathcal{F}$ are the *plaintext space*, the *ciphertext space*, the *label space*, and the *admissible function space*, respectively.

4.1 Syntax

Labeled Programs. First, let us define labeled programs, a concept first introduced in [19].

For each HAE, a set of admissible functions $\mathcal{F}$ is associated. In reality, an element f of $\mathcal{F}$ is a concrete representation of a function which can be evaluated in polynomial time. It is required that any $f \in \mathcal{F}$ should represent a function of form $f : \mathcal{M}^l \to \mathcal{M}$ for some $l \in \mathbb{Z}^+$ which depends on f. We will simply call an element $f \in \mathcal{F}$ an *admissible function*. The number l is the *arity* of f.

A HAE encrypts a plaintext $m \in \mathcal{M}$ under a 'label' $\tau \in \mathcal{L}$, and a labeled program is an admissible function together with information which plaintexts should be used as inputs. Formally, a *labeled program* is a tuple $P = (f, \tau_1, \ldots, \tau_l)$, where f is an arity-l admissible function, and $\tau_i \in \mathcal{L}$ are labels for $i = 1, \ldots, l$ for each input of f. The idea is that, if m_i are plaintexts encrypted under the label τ_i, respectively, then the evaluation of the labeled program $P = (f, \tau_1, \ldots, \tau_l)$ is $f(m_1, \ldots, m_l)$.

We also define the *identity labeled program* with label $\tau \in \mathcal{L}$, which is $I_\tau = (\mathsf{id}, \tau)$, where $\mathsf{id} : \mathcal{M} \to \mathcal{M}$ is the identity function.

Homomorphic Authenticated Encryption. A HAE $\mathit{\Pi}$ consists of the following four PPT algorithms.

- $\mathrm{Gen}(1^\lambda)$: given a security parameter λ, Gen outputs a key pair (ek, sk), with a public evaluation key ek and a secret key sk.
- $\mathrm{Enc}(sk, \tau, m)$: given a secret key sk, a label τ and a plaintext m, Enc outputs a ciphertext c.
- $\mathrm{Eval}(ek, f, c_1, \cdots, c_l)$: given an evaluation key ek, an admissible function $f : \mathcal{M}^l \to \mathcal{M}$ and l ciphertexts $c_1, \cdots, c_l \in \mathcal{C}$, the deterministic algorithm Eval outputs a ciphertext $\tilde{c} \in \mathcal{C}$.
- $\mathrm{Dec}(sk, (f, \tau_1, \cdots, \tau_l), \hat{c})$: when given a secret key sk, a labeled program $(f, \tau_1, \cdots, \tau_l)$ and a ciphertext $\hat{c} \in \mathcal{C}$, the deterministic algorithm Dec outputs a message $m \in \mathcal{M}$ or $\perp$.

We assume that ek implicitly contains the information about $\mathcal{M}$, $\mathcal{C}$, $\mathcal{L}$, and $\mathcal{F}$. As mentioned above, we assume both Eval and Dec are deterministic algorithms.

Compactness. In order to exclude trivial constructions, we require that the output size of $\mathrm{Eval}(ek, \dots)$ and $\mathrm{Enc}(sk, \cdot, \cdot)$ should be bounded by a polynomial of λ for any choice of their input, when $(ek, sk) \leftarrow \mathrm{Gen}(1^\lambda)$. That is, the ciphertext size is independent of the choice of the admissible function f.

Correctness. A HAE scheme must satisfy the following correctness properties:

- $m = \mathrm{Dec}(sk, I_\tau, \mathrm{Enc}(sk, \tau, m))$, for any $\lambda \in \mathbb{Z}^+$, $\tau \in \mathcal{L}$ and $m \in \mathcal{M}$, when $(ek, sk) \leftarrow \mathrm{Gen}(1^\lambda)$.
- $f(m_1, \ldots, m_l) = \mathrm{Dec}(sk, (f, \tau_1, \ldots, \tau_l), c)$, for any $\lambda \in \mathbb{Z}^+$, any $f \in \mathcal{F}$, any $\tau_i \in \mathcal{L}$, $m_i \in \mathcal{M}$ for $i = 1, \ldots, l$, when $(ek, sk) \leftarrow \mathrm{Gen}(1^\lambda)$, $c_i \leftarrow \mathrm{Enc}(sk, \tau_i, m_i)$ for $i = 1, \ldots, l$, and $c \leftarrow \mathrm{Eval}(ek, f, c_1, \ldots, c_l)$.

In addition, we require that a HAE should satisfy a property we call ciphertext constant testability, which will be explained in Sect. 4.3.

4.2 Legal Encryption

As in the case of homomorphic MAC [19], it is required that a label τ used in an encryption $\mathrm{Enc}(sk, \tau, m)$ should *not* be reused. In practice, this should be enforced by policy between valid users of HAE.

In the security model of HAE, this is expressed by legality of an encryption query of an adversary: the adversary makes adaptive encryption queries (τ, m), and this query is answered by $c \leftarrow \mathrm{Enc}(sk, \tau, m)$. Let us keep an encryption history S of all tuples (τ, m, c) occurring as the result of such an encryption query. We say that an encryption query (τ, m) is *illegal*, if $(\tau, \cdot, \cdot) \in S$. Also, we say that τ is *new* if $(\tau, \cdot, \cdot) \notin S$, and τ is *used* if $(\tau, \cdot, \cdot) \in S$. We say that an adversary is *legal*, if it does not make any illegal queries, including illegal encryption queries.

In this paper, we only consider legal adversaries, that is, we always *exclude* illegal adversaries, in the sense of the paper by Bellare, Hofheinz, and Kiltz [3]. Different security notions may have additional definitions of illegal queries (for example, illegal decryption queries), but any encryption query involving a used label will always be considered as illegal.

4.3 Constant Testability

For later use, we need to be able to check efficiently whether certain functions are constant or not. For example, we need this for the homomorphic evaluation of any admissible function f, regarded as a function mapping a ciphertext tuple to a ciphertext: $(c_1, \ldots, c_l) \mapsto \mathrm{Eval}(ek, f, c_1, \ldots, c_l)$. In fact, we need to consider slightly more general functions.

Fix a HAE Π and an evaluation key ek of Π. Given an arity-l admissible function f, a subset I of the index set $\{1, \cdots, l\}$, plaintexts $(m_i)_{i \in I} \in \mathcal{M}^{|I|}$, and ciphertexts $(c_i)_{i \in I} \in \mathcal{C}^{|I|}$, we make the following definition:

Definition 3. *A* partial application *of f w.r.t. plaintexts $(m_i)_{i \in I}$ is the function $\tilde{f} : \mathcal{M}^{l-|I|} \to \mathcal{M}$ defined by $\tilde{f}((m_j)_{j \notin I}) := f(m_1, \ldots, m_l)$. We denote this $\tilde{f}$ by $\mathsf{App}(f, (m_i)_{i \in I})$.*

Definition 4. *A* partial homomorphic evaluation *of f w.r.t. ciphertexts $(c_i)_{i \in I}$ is the function $\tilde{e} : \mathcal{C}^{l-|I|} \to \mathcal{C}$ defined by $\tilde{e}((c_j)_{j \notin I}) := \mathrm{Eval}(ek, f, c_1, \ldots, c_l)$. We denote this $\tilde{e}$ by $\mathsf{Eval}(f, (c_i)_{i \in I})$.*

So, $\tilde{f} = \mathsf{App}(f, (m_i)_{i \in I})$ is the admissible function f with some inputs m_i for $i \in I$ already 'filled in', and $\tilde{f}$ becomes a function of remaining plaintext inputs. Similarly, $\tilde{e} = \mathsf{Eval}(f, (c_i)_{i \in I})$ is the homomorphic evaluation $\mathrm{Eval}(ek, f, c_1, \ldots, c_l)$ with some inputs c_i for $i \in I$ already filled in, and $\tilde{e}$ becomes a function of remaining ciphertext inputs. In particular, $\mathsf{Eval}(f, (c_i)_{i \in I})$ is a constant function if $I = \{1, \ldots, l\}$.

Informally, if a HAE Π satisfies *ciphertext constant testability* (CCT), then there is an efficient algorithm which can determine whether such $\mathsf{Eval}(f, (c_i)_{i \in I})$ is constant or not with overwhelming probability.

In fact, we need a computational version of this property. Therefore, we formally define CCT using the following security game $\mathrm{CCT}_{\Pi, A, D}$ involving an adversary A and a 'constant tester' D:

$\mathbf{CCT}_{\Pi, A, D}(1^\lambda)$:

Initialization. A key pair $(ek, sk) \leftarrow \mathrm{Gen}(1^\lambda)$ is generated, a set S is initialized as the empty set $\emptyset$. Then ek is given to A.

Queries. A may make legal encryption queries adaptively. For each encryption query (τ, m) of A, the answer $c \leftarrow \mathrm{Enc}(sk, \tau, m)$ is returned to A, and S is updated as $S \leftarrow S \cup \{(\tau, m, c)\}$.

Challenge. A outputs a labeled program $(f, \tau_1, \ldots, \tau_l)$. Let I be the set of indices $i = 1, \ldots, l$ such that $(\tau_i, m_i, c_i) \in S$ for some[2] m_i, c_i. Then D outputs a bit $b \leftarrow D(ek, (f, \tau_1, \ldots, \tau_l), I, (c_i)_{i \in I})$.

Finalization. The game outputs 1 if $\tilde{e} := \mathsf{Eval}(f, (c_i)_{i \in I})$ is constant and $b = 0$, or $\tilde{e}$ is nonconstant and $b = 1$. The game outputs 0 otherwise.

The output bit b of the tester D is 1 iff D 'thinks' that $\mathsf{Eval}(f, (c_i)_{i \in I})$ is constant. Therefore, the event $\mathrm{CCT}_{\Pi,A,D}(1^\lambda) = 1$ happens when the tester D is wrong.

The advantage of an adversary A in the game CCT against D is defined as

$$\mathbf{Adv}^{\mathrm{CCT}}_{\Pi,A,D}(\lambda) := \Pr[\mathrm{CCT}_{\Pi,A,D}(1^\lambda) = 1] \ .$$

We say that a HAE Π satisfies the *ciphertext constant testability (CCT)*, if there exists a PPT constant tester D such that $\mathbf{Adv}^{\mathrm{CCT}}_{\Pi,A,D}(\lambda)$ is negligible for any *legal* PPT adversary A.

Similarly, we may define *plaintext constant testability* (PCT): informally, Π satisfies PCT if testing whether a partial application of an admissible function is constant or not can be done efficiently.

When the set of admissible functions of a HAE is simple, both PCT and CCT may be satisfied. But, PCT might be a difficult property to be satisfied in general; if a HAE supports boolean circuits and is fully homomorphic, then since satisfiability of a boolean circuit can be efficiently determined if constant testing is efficient, if such HAE satisfies PCT, we may use it to invert any one-way function.

On the other hand, we claim that a HAE to satisfy CCT is a relatively mild requirement: unlike the plaintext space $\mathcal{M}$, often the ciphertext space $\mathcal{C}$ might be a large ring, and $\mathsf{Eval}(f, (c_i)_{i \in I})$ is a polynomial on the ring $\mathcal{C}$, in which case we may use the Schwartz-Zippel lemma to perform polynomial identity testing. This applies to our HAE scheme to be presented in this paper, as shown in Theorem 5.

Moreover, we show that if Π is a HAE which does not necessarily satisfy CCT, then there is a simple generic transformation which turns it into another HAE Π' which satisfies CCT, while preserving original security properties satisfied by Π. This will be explained in Sect. 4.7.

Therefore, without (much) loss of generality, we assume the CCT property to be an additional requirement for a HAE to satisfy.

4.4 Privacy

Indistinguishability under Chosen-Plaintext Attack. First we define a homomorphic version of the IND-CPA security based on the left-or-right indistinguishability [1]. We use the following security game $\text{IND-CPA}_{\Pi,A}$ for an adversary A:

[2] For any $i \in I$, such m_i, c_i are necessarily unique.

IND-CPA$_{\Pi,A}(1^\lambda)$:

Initialization. A key pair $(ek, sk) \leftarrow \mathrm{Gen}(1^\lambda)$ is generated, a set S is initialized as the empty set $\emptyset$. And a coin $b \overset{\$}{\leftarrow} \{0,1\}$ is flipped. Then ek is given to A.

Queries. A may make encryption queries adaptively. For each encryption query (τ, m_0, m_1) of A, the answer $c \leftarrow \mathrm{Enc}(sk, \tau, m_b)$ is returned to A, and S is updated as $S \leftarrow S \cup \{(\tau, (m_0, m_1), c)\}$.

Finalization. A outputs a bit b', and then the challenger returns 1 if $b = b'$, and 0 otherwise.

As usual, an encryption query is considered illegal if it involves a used label.[3] The advantage of A in the game IND-CPA for the scheme Π is defined by

$$\mathbf{Adv}^{\text{IND-CPA}}_{\Pi,A}(\lambda) := \left| \Pr[\,\text{IND-CPA}_{\Pi,A}(1^\lambda) = 1] - \frac{1}{2} \right| \ .$$

We say that Π satisfies IND-CPA, if $\mathbf{Adv}^{\text{IND-CPA}}_{\Pi,A}(\lambda)$ is negligible for any *legal* PPT adversary A.

Indistinguishability under Chosen-Ciphertext attack. Though the usual IND-CCA security is not achievable for homomorphic encryption due to malleability, nevertheless we may define a version of IND-CCA for HAE. It is because that for HAE, a ciphertext is decrypted with respect to a labeled program; while the ciphertext is still malleable by function evaluation, a decryption query should essentially declare how it was produced. This allows a homomorphic version of IND-CCA to be defined naturally as follows.

The most important difference of our definition is on the legality of a decryption query. In our case, any decryption query for a ciphertext produced by function evaluation which may nontrivially depend on the bit b should be considered illegal, since decryption of that ciphertext might reveal the bit b. To formalize:

Let S be the encryption history as before. Then, we say that a decryption query $((f, \tau_1, \cdots, \tau_l), \hat{c})$ is *illegal*, if $\tilde{f}_0$ and $\tilde{f}_1$ are not equal, where

$$I := \{i \in \{1, \cdots, l\} \mid (\tau_i, (m_{i,0}, m_{i,1}), \cdot) \in S \text{ for some } (m_{i,0}, m_{i,1}) \in \mathcal{M} \times \mathcal{M}\} \ ,$$
$$\tilde{f}_b := \mathsf{App}(f, (m_{i,b})_{i \in I}) \text{ for } b = 0, 1 \ .$$

Homomorphic IND-CCA for a HAE $\Pi = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Eval}, \mathrm{Dec})$ is defined via the security game IND-CCA$_{\Pi,A}$ whose formal description we omit here due to the space constraints. It is very similar to IND-CPA$_{\Pi,A}$, except that the adversary A may make both encryption and decryption queries at any time. In the full version of this paper, the formal description of the security game IND-CCA$_{\Pi,A}$ will be given, as well as other security games.

[3] As before, a label τ is used if $(\tau, \cdot, \cdot) \in S$

The advantage of A in the game IND-CCA for the scheme Π, $\mathbf{Adv}_{\Pi,A}^{\text{IND-CCA}}(\lambda)$, is defined similarly. And we say that Π satisfies IND-CCA, if $\mathbf{Adv}_{\Pi,A}^{\text{IND-CCA}}(\lambda)$ is negligible for any *legal* PPT adversary A.

Remark 1. As we have discussed while defining constant testability in Sect. 4.3, in general it may not be feasible to check whether $\tilde{f}_0 = \tilde{f}_1$ or not, especially when the HAE in question is fully homomorphic and may process arbitrarily large boolean circuits. Therefore, in general, it may not be feasible to efficiently decide whether a decryption query is legal or not. Hence we use the exclusion-style definition, rather than the penalty-style, according to the classification of Bellare, Hofheinz, and Kiltz [3]: we regard only legal adversaries, which does not make any illegal queries. While in other cases the two styles of definitions are mostly compatible, in this case it is not.

Also, later in Sect. 4.7, we show that by using a secure PRF and a collision-resistant hash function (or a hash tree), we may transform a HAE scheme into another HAE which satisfies CCT. After we apply the transformation, if $\mathrm{Dec}(sk, (f, \tau_1, \ldots, \tau_l), \hat{c}) \neq \perp$, then with overwhelming probability we should have $I = \{1, \ldots, l\}$. Therefore, any decryption query will either output $\perp$, or both $\tilde{f}_0$ and $\tilde{f}_1$ are constant, which makes deciding if a verification query is illegal trivial. This transform can be used if an application requires ability to efficiently decide whether a verification query is illegal or not.

4.5 Authenticity

Unforgeability under Chosen-Plaintext Attack. Our authenticity definition for HAE is an adaptation of the definition given by Catalano and Fiore [11] for homomorphic MACs.

First, we define the forgery of an adversary. Let $((f, \tau_1, \cdots, \tau_l), \hat{c})$ be a forgery attempt, and let S be the encryption history. We say that it is a *forgery*, if the following holds:

1. It is *valid*, that is, $\perp \neq \mathrm{Dec}(sk, (f, \tau_1, \cdots, \tau_l), \hat{c})$ and,
2. One of the following holds:
   - Type 1 forgery: $\mathsf{App}(f, (m_i)_{i \in I})$ is not constant, or,
   - Type 2 forgery: $\mathsf{App}(f, (m_i)_{i \in I})$ is constantly equal to some $\tilde{m}$, but $\tilde{m} \neq \mathrm{Dec}(sk, (f, \tau_1, \cdots, \tau_l), \hat{c})$,

   where I is the set of $i \in \{1, \ldots, l\}$ such that $(\tau_i, m_i, \cdot) \in S$ for some (unique) $m_i \in \mathcal{M}$.

We define the *unforgeability under chosen-plaintext attack* (UF-CPA) of a HAE Π using the security game $\text{UF-CPA}_{\Pi,A}$. In the game, the adversary A is given the evaluation key ek and the encryption oracle. Finally, A outputs $((f, \tau_1, \cdots, \tau_l), \hat{c})$. The game outputs 1 if it is a successful forgery, and 0 otherwise.

The advantage of A in the game UF-CPA for the scheme Π is defined as

$$\mathbf{Adv}_{\Pi,A}^{\text{UF-CPA}}(\lambda) := \Pr[\,\text{UF-CPA}_{\Pi,A}(1^\lambda) = 1] \ .$$

We say that Π satisfies UF-CPA, if $\mathbf{Adv}_{\Pi,A}^{\text{UF-CPA}}(\lambda)$ is negligible for any *legal* PPT adversary A.

Unforgeability under Chosen-Ciphertext attack. It is also natural to consider a stronger variant of unforgeability, in which an adversary is allowed to make decryption queries as well as encryption queries. We call this variant UF-CCA. The only difference of UF-CCA from UF-CPA is that the adversary A can also make any decryption query $((f, \tau_1, \cdots, \tau_l), \hat{c})$, which is answered with $\text{Dec}(sk, (f, \tau_1, \cdots, \tau_l), \hat{c})$.

The advantage of A in the game UF-CCA for the scheme Π, $\mathbf{Adv}_{\Pi,A}^{\text{UF-CCA}}(\lambda)$, is defined similarly. And we say that Π satisfies UF-CCA, if $\mathbf{Adv}_{\Pi,A}^{\text{UF-CCA}}(\lambda)$ is negligible for any *legal* PPT adversary A.

Strong Unforgeability under Chosen-Plaintext Attack. Sometimes it is useful to consider stronger definition of authenticity. So let us define *strong unforgeability* for HAE. Let S be the encryption history. Then we say that a forgery attempt $((f, \tau_1, \cdots, \tau_l), \hat{c})$ is a *strong forgery*, if the following holds:

1. It is valid, that is, $\perp \neq \text{Dec}(sk, (f, \tau_1, \cdots, \tau_l), \hat{c})$ and,
2. One of the following holds:
   - Type 1 strong forgery: $\mathsf{Eval}(f, (c_i)_{i\in I})$ is not constant, or,
   - Type 2 strong forgery: $\mathsf{Eval}(f, (c_i)_{i\in I})$ is constantly equal to some $\tilde{c}$, but $\tilde{c} \neq \hat{c}$, where I is the set of $i \in \{1, \ldots, l\}$ such that $(\tau_i, \cdot, c_i) \in S$ for some (unique) $c_i \in \mathcal{C}$.

We define the *strong unforgeability under chosen-plaintext attack* (SUF-CPA) of a HAE Π using the game $\text{SUF-CPA}_{\Pi,A}$. In the game, the adversary A is given the evaluation key ek and the encryption oracle. Finally, A outputs $((f, \tau_1, \cdots, \tau_l), \hat{c})$. The game outputs 1 iff it is a successful strong forgery.

The advantage of A in the game SUF-CPA for the scheme Π, $\mathbf{Adv}_{\Pi,A}^{\text{SUF-CPA}}(\lambda)$, is defined similarly. And we say that Π satisfies SUF-CPA, if $\mathbf{Adv}_{\Pi,A}^{\text{SUF-CPA}}(\lambda)$ is negligible for any *legal* PPT adversary A.

Strong Unforgeability under Chosen-Ciphertext Attack. Also for strong unforgeability, we consider security against chosen-ciphertext attacks, which we call SUF-CCA. Again, the only difference of SUF-CCA from SUF-CPA is that the adversary A is also given the decryption oracle.

The advantage of A in the game SUF-CCA for the scheme Π, $\mathbf{Adv}_{\Pi,A}^{\text{SUF-CCA}}(\lambda)$, is defined similarly. And we say that Π satisfies SUF-CCA, if $\mathbf{Adv}_{\Pi,A}^{\text{SUF-CCA}}(\lambda)$ is negligible for any *legal* PPT adversary A.

4.6 Relations on Security Notions

In this section, we investigate relations between the six security notions defined in the previous section. First, we have trivial implications from CCA security to CPA security.

Theorem 1. *UF-CCA implies UF-CPA, SUF-CCA implies SUF-CPA, and also IND-CCA implies IND-CPA.*

The following theorem says that the strong unforgeability implies unforgeability.

Theorem 2. *SUF-CCA implies UF-CCA. And SUF-CPA implies UF-CPA.*

Proof. It is enough to show that a successful forgery is also a successful strong forgery. Let $((f, \tau_1, \cdots, \tau_l), \hat{c})$ be a forgery. If it is a type 1 forgery, then $\tilde{f} := \mathsf{App}(f, (m_i)_{i \in I})$ is not constant. That is, there exist two tuples $(m_j^1)_{j \notin I}$ and $(m_j^2)_{j \notin I}$ such that $\tilde{f}(m_j^1)_{j \notin I} \neq \tilde{f}(m_j^2)_{j \notin I}$. Then there exist two tuples $(c_j^1)_{j \notin I}$ and $(c_j^2)_{j \notin I}$ such that $m_j^1 = \mathrm{Dec}(sk, I_{\tau_j}, c_j^1)$ and $m_j^2 = \mathrm{Dec}(sk, I_{\tau_j}, c_j^2)$ for each $j \notin I$. Then we can show that $\tilde{e} := \mathsf{Eval}(f, (c_i)_{i \in I})$ is nonconstant; since we have $\mathrm{Dec}(sk, (f, \tau_1, \ldots, \tau_l), \tilde{e}(c_j^b)_{j \notin I}) = \tilde{f}(m_j^b)_{j \notin I}$ for $b = 1, 2$ by correctness, we see that $\tilde{e}(c_j^1)_{j \notin I} \neq \tilde{e}(c_j^2)_{j \notin I}$. So it is a type 1 strong forgery.

If it is a type 2 forgery but not a type 1 strong forgery, then both $\tilde{f}$ and $\tilde{e}$ are constants. Let the constant value of $\tilde{f}$ be $\tilde{m} \in \mathcal{M}$, and the constant value of $\tilde{e}$ be $\tilde{c} \in \mathcal{C}$. We have $\tilde{m} \neq \mathrm{Dec}(sk, (f, \tau_1, \cdots, \tau_l), \hat{c})$. But $\tilde{m} = \mathrm{Dec}(sk, (f, \tau_1, \cdots, \tau_l), \tilde{c})$, again by correctness. So $\hat{c} \neq \tilde{c}$, and thus it is a type 2 strong forgery.

Bellare et al. [2] showed that, in case of MAC, strong unforgeability implies strong unforgeability even when the adversary has access to the verification oracle, and in case of AE, integrity of ciphertexts implies integrity of ciphertexts even when the adversary has access to the verification oracle. The following can be considered as a homomorphic analogue to the result.

Theorem 3. *SUF-CPA implies SUF-CCA.*

The basic intuition of the proof of Theorem 3 is as follows: if a HAE scheme Π satisfies SUF-CPA, since it is infeasible to produce any strong forgery, essentially any decryption query $((f, \tau_1, \cdots, \tau_l), \hat{c})$ which should be answered with anything other than $\perp$ must be the output of the Eval algorithm with correct ciphertexts from encryption queries as inputs. Therefore, even if the decryption oracle is given to the adversary A, it would not give any useful information. The actual proof, which will be on the full version of this paper due to the page constraints, uses a hybrid argument where the decryption queries are in the end handled by a decryption simulation.

Theorem 4. *IND-CPA and SUF-CPA together imply IND-CCA.*

Proof of this theorem is similar to that of Theorem 3. Again, we use a hybrid argument to transform the IND-CCA game into another game that is essentially same as the IND-CPA game: the strong unforgeability allows us to simulate decryption oracle. Again, the complete proof will be given in the full version of this paper.

In conclusion, we see that IND-CPA and SUF-CPA together imply the strongest security notions, IND-CCA and SUF-CCA.

4.7 Generic Transformation for Ciphertext Constant Testability

Suppose that Π is a HAE which does *not* necessarily satisfy CCT. We describe a generic construction that transforms a HAE Π into another HAE Π' satisfying CCT, while preserving IND-CPA or SUF-CPA. The construction uses a PRF $F_k : \{0,1\}^\lambda \to \{0,1\}^\lambda$ and a family $\mathcal{H}$ of collision-resistant hash functions $H : \{0,1\}^* \to \{0,1\}^\lambda$.

Scheme $\Pi' = (\mathrm{Gen}', \mathrm{Enc}', \mathrm{Eval}', \mathrm{Dec}')$:

- $\mathrm{Gen}'(1^\lambda)$: Generate keys $(ek, sk) \leftarrow \mathrm{Gen}(1^\lambda)$, $k \leftarrow \{0,1\}^\lambda$ and $H \leftarrow \mathcal{H}$. Return (ek', sk') where $ek' = (ek, H)$ and $sk' = (sk, k)$.
- $\mathrm{Enc}'(sk', \tau, m)$: Let $h = F_k(\tau)$ and $c \leftarrow \mathrm{Enc}(sk, \tau, m)$. Return $c' = (h, c)$.
- $\mathrm{Eval}'(ek', f, c'_1, \cdots, c'_l)$: Parse $c'_i = (h_i, c_i)$ for $i = 1, \cdots, l$. Let $\tilde{h} = H(h_1, \cdots, h_l)$ and $\tilde{c} \leftarrow \mathrm{Eval}(ek, f, c_1, \cdots, c_l)$. Return $\tilde{c}' = (\tilde{h}, \tilde{c})$.
- $\mathrm{Dec}'(sk', (f, \tau_1, \cdots, \tau_l), \tilde{c}')$: Parse $\tilde{c}' = (\tilde{h}, \tilde{c})$. For each $i = 1, \cdots, l$, let $h_i = F_k(\tau_i)$. If $\tilde{h} = H(h_1, \cdots, h_l)$, then return $\mathrm{Dec}(sk, (f, \tau_1, \cdots, \tau_l), \tilde{c})$. Otherwise, return $\perp$.

It is clear that Π' satisfies correctness properties, as long as Π also does.

We claim that in addition Π' satisfies CCT. The constant tester D for Π' is simple: given ek', $(f, \tau_1, \ldots, \tau_l)$, I, and $(c'_i)_{i \in I}$, the tester D outputs 1 if $I = \{1, \ldots, l\}$, and outputs 0 otherwise. Suppose there exists an adversary A with non-negligible advantage in the game CCT against this tester. Observe that D errs only when $I \neq \{1, \ldots, l\}$ and the function $\mathsf{Eval}(f, (c'_i)_{i \in I})$ is constant. Therefore, we may use A to construct a hash collision finding algorithm B as follows: B receives $H \leftarrow \mathcal{H}$, and simulates the CCT game. Since B itself generates (ek, sk) and k, it may answer any queries made by A. Eventually A outputs a labeled program $(f, \tau_1, \ldots, \tau_l)$. If all τ_i are used, then B aborts. But there is a non-negligible probability that $I \neq \{1, \ldots, l\}$ and $\mathsf{Eval}(f, (c'_i)_{i \in I})$ is constant, and this means that $\tilde{h} = H(h_1, \ldots, h_l)$ as a function of $(h_j)_{j \notin I}$ is also constant. Therefore, B may output a collision pair with non-negligible probability, because $\{1, \ldots, l\} \setminus I \neq \emptyset$ and B may arbitrarily choose $h_j \neq h'_j$ for $j \notin I$.

Also, if Π satisfies IND-CPA then so does Π': informally, in the ciphertext $c' = (h, c)$ the h-part $F_k(\tau)$ has no information about the plaintext m, and any information about the plaintext m in the c-part $\mathrm{Enc}(sk, \tau, m)$ is computationally hidden since Π is IND-CPA.

And, if Π satisfies SUF-CPA, then so does Π': since any strong forgery $((f, \tau_1, \ldots, \tau_l), \hat{c}' = (\hat{h}, \hat{c}))$ of Π' has to be valid, it is easy to see that with negligible exception, all τ_i are used and $\mathsf{Eval}(f, (c'_i)_{i \in I})$ is constant. Then we may show that in fact $((f, \tau_1, \ldots, \tau_l), \hat{c})$ should be a type 2 strong forgery for Π.

We will provide proofs for all of the above in the full version of this paper. Note that one disadvantage of this transform is that Π' does not support composition of admissible functions; if Π supports boolean circuits, is fully homomorphic, and admissible functions are composable, then, we may want the same for Π'. For this, we may adopt the Merkle hash tree construction used by

Gennaro and Wichs [19] for their fully homomorphic MAC. This hash tree based transformation will also be given in the full version.

5 Construction

Here we describe our HAE Π and show that it satisfies correctness and CCT.

Parameters ρ, η, γ, $\bar{d}$ are polynomially bounded functions of the security parameter λ, and the modulus parameter Q is a function of λ satisfying $2 \leq Q \leq 2^\lambda$. We assume that all these parameters can be efficiently computed, given λ. Constraints on these parameters are given after the description of the scheme.

We use a PRF F in our construction. We may assume that $F_k : \{0,1\}^\lambda \to \mathbb{Z}_{q_0}$ for each $k \in \{0,1\}^\lambda$. The message space and the ciphertext space of our scheme is $\mathbb{Z}_Q$ and $\mathbb{Z}_{y_0}$, resp., and the label space is $\{0,1\}^\lambda$. To represent admissible functions we use arithmetic circuits, that is, circuits consisting of + gates and × gates. Such a circuit f of arity l determines a polynomial $f : \mathbb{Z}^l \to \mathbb{Z}$ with integral coefficients. We use such a circuit to compute function values of plaintext inputs in $\mathbb{Z}_Q$, and also to homomorphically evaluate ciphertexts in $\mathbb{Z}_{y_0}$. The precise description of the admissible function space will be given after the scheme description, together with discussions on the correctness property.

Scheme $\Pi = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Eval}, \mathrm{Dec})$:

- $\mathrm{Gen}(1^\lambda)$: Choose $p \overset{\$}{\leftarrow} [2^{\eta-1}, 2^\eta) \cap \mathrm{PRIME}$, $q_0 \overset{\$}{\leftarrow} [0, 2^\gamma/p) \cap \mathrm{ROUGH}(2^{\lambda^2})$, and $k \leftarrow \{0,1\}^\lambda$. Let $y_0 = pq_0$. Return the key pair (ek, sk), where $ek = (1^\lambda, y_0)$, and $sk = (1^\lambda, p, q_0, k)$.
- $\mathrm{Enc}(sk, \tau, m)$: Given the secret key sk, a label $\tau \in \{0,1\}^\lambda$ and a plaintext $m \in \mathbb{Z}_Q$, choose $r \overset{\$}{\leftarrow} \mathbb{Z} \cap (-2^\rho, 2^\rho)$. Let $a = rQ + m$ and $b = F_k(\tau)$. Return $c = \mathrm{CRT}_{(p,q_0)}(a, b)$.
- $\mathrm{Eval}(ek, f, c_1, \cdots, c_l)$: Given ek, an arithmetic circuit f of arity l and ciphertexts $c_1, \cdots, c_l$, return $\tilde{c} := f(c_1, \cdots, c_l) \bmod y_0$
- $\mathrm{Dec}(sk, (f, \tau_1, \cdots, \tau_l), \hat{c})$: For $i = 1$ to l, compute $b_i \leftarrow F_k(\tau_i)$ and $b = f(b_1, \cdots, b_l) \bmod q_0$. Return $m = (\hat{c} \bmod p) \bmod Q$, if $b = \hat{c} \bmod q_0$. Otherwise, return $\perp$.

5.1 Correctness

Here we determine when our HAE scheme is correct. Let $(ek, sk) \leftarrow \mathrm{Gen}(1^\lambda)$. Let $c_i \leftarrow \mathrm{Enc}(sk, \tau_i, m_i)$ for each $i = 1, \cdots, l$. And let $\tilde{c} \leftarrow \mathrm{Eval}(ek, f, c_1, \cdots, c_l)$, for any arity-$l$ arithmetic circuit f of degree d. Then

$$\begin{aligned}\tilde{c} \bmod p &= (f(c_1, \cdots, c_l) \bmod y_0) \bmod p = f(c_1, \cdots, c_l) \bmod p \\ &= f(c_1 \bmod p, \cdots, c_l \bmod p) \bmod p \\ &= f(r_1 Q + m_1, \cdots, r_l Q + m_l) \bmod p \\ &= f(r_1 Q + m_1, \cdots, r_l Q + m_l)\end{aligned}$$

The last equality in the above holds if $|\, f(r_1Q + m_1, \cdots, r_lQ + m_l)\,| \le p/2$. And so, in this case,

$$\begin{aligned}(\tilde{c} \bmod p) \bmod Q &= f(r_1Q + m_1, \cdots, r_lQ + m_l) \bmod Q \\ &= f(m_1, \cdots, m_l) \bmod Q \ .\end{aligned}$$

Since $|\, f(r_1Q + m_1, \cdots, r_lQ + m_l)\,| \le \|f\|_1 \cdot 2^{d(\rho+\lambda)}$ and $2^{\eta-2} \le p/2$, the correctness is guaranteed if $\|f\|_1 \cdot 2^{d(\rho+\lambda)} \le 2^{\eta-2}$, where $\|f\|_1$ is the ℓ_1-norm of the coefficient vector of f.

So, we can see that if $\|f\|_1 \le 2^{\eta/2}$ and $\eta \ge 2d(\rho + \lambda) + 4$, then the correct decryption is guaranteed for $\tilde{c}$. Let $\bar{d}$ be the parameter representing the maximum degree for our admissible functions. Then, as long as the condition $\eta \ge 2\bar{d}(\rho + \lambda) + 4$ is met, we may define an admissible function as an arithmetic circuit f such that $\deg f \le \bar{d}$ and $\|f\|_1 \le 2^{\eta/2}$ as a polynomial.

5.2 Constraints of the Parameters

In our scheme, the parameters must satisfy the following constraints:

- $\rho = \omega(\lg \lambda)$: to resist the brute force attack on the EF-AGCD problem.
- $\eta \ge 2\bar{d}(\rho + \lambda) + 4$: for the correctness.
- $\eta \ge \Omega(\lambda^2)$: to resist the factoring attack using the elliptic curve method (ECM). In fact, we also want $\eta \ge \lambda^2 + 1$ to make y_0 a 2^{λ^2}-rough integer.
- $\gamma = \eta^2\omega(\lg \lambda)$: to resist known attacks on the EF-AGCD problem as explained in [17, 13].
- $2 \le Q \le 2^{\lambda}$: to ensure that $\gcd(Q, y_0) = 1$.

Assuming $\bar{d} = \Theta(\lambda)$, one possible choice of parameters which satisfies all of above is $\rho = \Theta(\lambda)$, $\eta = \Theta(\lambda^2)$, and $\gamma = \Theta(\lambda^5)$.

5.3 Ciphertext Constant Testability

Theorem 5. *The scheme Π satisfies CCT.*

Proof. Let ek be an evaluation key generated by $\mathrm{Gen}(1^{\lambda})$, f be any admissible arity-l arithmetic circuit, and $(c_i)_{i\in I}$ be any element in $\mathbb{Z}_{y_0}^{|\,I\,|}$ for a subset I of $\{1, \cdots, l\}$.

The constant tester D for our scheme Π determines if $\tilde{e} := \mathsf{Eval}(f, (c_i)_{i\in I})$ is constant or not with overwhelming probability, as follows: given ek, $(f, \tau_1, \ldots, \tau_l)$, I, and $(c_i)_{i\in I}$, the tester D outputs 1 if $I = \{1, \ldots, l\}$. Otherwise, it samples two tuples of ciphertexts $(c_j^0)_{j\notin I}, (c_j^1)_{j\notin I} \overset{\$}{\leftarrow} (\mathbb{Z}_{y_0})^{l-|\,I\,|}$. Finally, if $\tilde{e}(c_j^0)_{j\notin I} \equiv \tilde{e}(c_j^1)_{j\notin I} \pmod{y_0}$, then D outputs 1, and otherwise D outputs 0.

The tester D is essentially doing the usual polynomial identity testing. In the scheme Π, $\tilde{e}$ can be considered as an $(l - |\,I\,|)$-variate polynomial over $\mathbb{Z}_{y_0}$ of degree $\le \deg f$. We have

$$\tilde{e}(c_j)_{j\notin I} = f((c_i)_{i\in I}, (c_j)_{j\notin I}) = f(c_1, \ldots, c_l) \bmod y_0 \ .$$

When $I = \{1, \cdots, l\}$, $\tilde{e}$ is clearly constant and D outputs 1 correctly. In case $I \neq \{1, \cdots, l\}$, consider the function $\tilde{e}' := \tilde{e} - \tilde{e}(c_j^0)_{j \notin I} \bmod y_0$ for $(c_j^0)_{j \notin I} \in (\mathbb{Z}_{y_0})^{l-|I|}$. If $\tilde{e}$ is constant, then $\tilde{e}'$ is constantly zero and $\tilde{e}'(c_j^1)_{j \notin I} = \tilde{e}(c_j^1)_{j \notin I} - \tilde{e}(c_j^0)_{j \notin I} \equiv 0 \pmod{y_0}$ for any $(c_j^1)_{j \notin I} \in (\mathbb{Z}_{y_0})^{l-|I|}$. So, $\tilde{e}(c_j^0)_{j \notin I} \equiv \tilde{e}(c_j^1)_{j \notin I} \pmod{y_0}$ and D outputs 1 correctly. If $\tilde{e}$ is not constant, then $\tilde{e}'$ is not constantly zero and D outputs the incorrect answer 1 when $\tilde{e}(c_j^0)_{j \notin I} \equiv \tilde{e}(c_j^1)_{j \notin I} \pmod{y_0}$, that is, $\tilde{e}'(c_j^1)_{j \notin I} \equiv 0 \pmod{y_0}$. This is the only case when D is incorrect. So the error probability of the tester D is

$$\Pr\left[\,\tilde{e}'(c_j^1)_{j \notin I} \equiv 0 \bmod y_0 \mid (c_j^1)_{j \notin I} \xleftarrow{\$} (\mathbb{Z}_{y_0})^{l-|I|}\right] ,$$

when $\tilde{e}'$ is not constantly zero.

Since y_0 is chosen as a 2^{λ^2}-rough random integer, with negligible exception, y_0 is square-free and $\tilde{e}'$ is not constantly zero modulo a prime factor $p' \geq 2^{\lambda^2}$ of y_0. Then, using Schwartz-Zippel lemma,

$$\begin{aligned}\Pr\left[\,\tilde{e}'(c_j^1)_{j \notin I} \equiv 0 \bmod y_0 \,|(c_j^1)_{j \notin I} \xleftarrow{\$} (\mathbb{Z}_{y_0})^{l-|I|}\right]& \\ \leq \Pr\left[\,\tilde{e}'(c_j^1)_{j \notin I} \equiv 0 \bmod p' \mid (c_j^1)_{j \notin I} \xleftarrow{\$} (\mathbb{Z}_{p'})^{l-|I|}\right]& \\ \leq \frac{\deg f}{p'} \leq \frac{\bar{d}}{2^{\lambda^2}} = negl(\lambda) \ .&\end{aligned}$$

Therefore, the error probability of the tester D is negligible.

6 Security

In this section, we prove our HAE scheme satisfies both IND-CPA and SUF-CPA. From this, we conclude that Π is IND-CCA and SUF-CCA by Theorem 3 and Theorem 4. For simplicity, we consider the scheme Π as an ideal scheme obtained by replacing the PRF F with a real random function from $\{0,1\}^\lambda$ into $\mathbb{Z}_{q_0}$. If F is secure, then the real scheme is secure if the ideal scheme is.

6.1 Privacy

As mentioned earlier, Coron et al. proved the equivalence of the EF-AGCD and the decisional EF-AGCD in [14]. So, Theorem 6 actually says that Π is IND-CPA under the EF-AGCD assumption.

Theorem 6. *The scheme Π is IND-CPA under the decisional (ρ, η, γ)-EF-AGCD assumption.*

Proof. We prove this theorem by a hybrid argument to transform the game IND-CPA into another game that is infeasible to break.

Let A be a PPT adversary engaging in the game IND-CPA. Without loss of generality, we assume that A makes exactly $q = q(\lambda)$ encryption queries. For each $i \in \{0, \ldots, q\}$, define IND-CPA^i to be the game that is equal to IND-CPA except that the first i encryption queries are answered by a sample from the uniform distribution over the ciphertext space $\mathbb{Z}_{y_0}$.

By definition, IND-CPA^0 = IND-CPA. So,

$$\mathbf{Adv}_{\Pi,A}^{\text{IND-CPA}^0}(\lambda) = \mathbf{Adv}_{\Pi,A}^{\text{IND-CPA}}(\lambda) \ ,$$

And the game IND-CPA^q does not reveal any information about the randomly chosen bit b. So,

$$\mathbf{Adv}_{\Pi,A}^{\text{IND-CPA}^q}(\lambda) = 0 \ .$$

Now consider the difference of each consecutive two games. We want to show that for each $i \in \{1, \ldots, q\}$, the difference between $\mathbf{Adv}_{\Pi,A}^{\text{IND-CPA}^{i-1}}(\lambda)$ and $\mathbf{Adv}_{\Pi,A}^{\text{IND-CPA}^i}(\lambda)$ is not greater than the advantage for the the decisional (ρ, η, γ)-EF-AGCD problem. For this purpose, we construct a PPT distinguisher $D(1^\lambda, y_0, \mathcal{D}(p, q_0, \rho), z)$ for the decisional (ρ, η, γ)-EF-AGCD problem as follows: D starts the simulation of the game IND-CPA$_{\Pi,A}^{i-1}$ or IND-CPA$_{\Pi,A}^{i}$ giving y_0 as an evaluation key to A. And $b \xleftarrow{\$} \{0,1\}$. Let $(\tau, m_0, m_1) \in \{0,1\}^\lambda \times \mathbb{Z}_Q \times \mathbb{Z}_Q$ be the j-th encryption query of A. Then D replies A with $c := (xQ + m_b) \bmod y_0$, where x is chosen as below.

$$\begin{aligned} j \leq i-1 &\implies x \xleftarrow{\$} \mathbb{Z}_{y_0} \ , \\ j = i &\implies x = z \ , \\ j \geq i+1 &\implies x \leftarrow \mathcal{D}(p, q_0, \rho) \ . \end{aligned}$$

Finally, D returns b', which is the output of A.

Note that $\gcd(y_0, Q) = 1$ since y_0 is 2^{λ^2}-rough and $Q \leq 2^\lambda$. Consider the answer $c = (xQ + m_b) \bmod y_0$ produced by D for an encryption query. If $x \xleftarrow{\$} \mathbb{Z}_{y_0}$, then c is also uniformly distributed over y_0. And if $x \leftarrow \mathcal{D}(p, q_0, \rho)$, then the distribution of c is identical to the distribution of $\text{Enc}(sk, \tau, m_b)$. Therefore, if $z \xleftarrow{\$} \mathbb{Z}_{y_0}$, then D simulates the game IND-CPA^i. And if $z \leftarrow \mathcal{D}(p, q_0, \rho)$, then D simulates the game IND-CPA^{i-1}. So, the difference between $\mathbf{Adv}_{\Pi,A}^{\text{IND-CPA}^{i-1}}(\lambda)$ and $\mathbf{Adv}_{\Pi,A}^{\text{IND-CPA}^i}(\lambda)$ is not greater than the advantage of D for the the decisional (ρ, η, γ)-EF-AGCD problem, which is negligible by the decisional (ρ, η, γ)-EF-AGCD assumption. That is,

$$\left| \mathbf{Adv}_{\Pi,A}^{\text{IND-CPA}^{i-1}}(\lambda) - \mathbf{Adv}_{\Pi,A}^{\text{IND-CPA}^i}(\lambda) \right| = negl(\lambda) \ ,$$

for any $i \in \{1, \cdots, q\}$.

Hence,

$$\begin{aligned} \mathbf{Adv}_{\Pi,A}^{\text{IND-CPA}}(\lambda) &\leq \mathbf{Adv}_{\Pi,A}^{\text{IND-CPA}^q}(\lambda) + \sum_{i=1}^{q} \left| \mathbf{Adv}_{\Pi,A}^{\text{IND-CPA}^{i-1}}(\lambda) - \mathbf{Adv}_{\Pi,A}^{\text{IND-CPA}^i}(\lambda) \right| \\ &\leq 0 + q \cdot negl(\lambda) \\ &= negl(\lambda) \ . \end{aligned}$$

Consequently, $\mathbf{Adv}_{\Pi,A}^{\text{IND-CPA}}(\lambda)$ is negiligible for any PPT adversary A and Π is IND-CPA.

6.2 Authenticity

Theorem 7. *If the (ρ, η, γ)-EF-AGCD assumption holds, then the scheme Π is SUF-CPA.*

Proof. Suppose there exists a PPT adversary A for the game SUF-CPA such that

$$\Pr\big[\,\text{SUF-CPA}_{\Pi,A}(1^\lambda) = 1\big] \geq \epsilon(\lambda) \enspace ,$$

for some non-negligible function $\epsilon > 0$.

Then, we construct a PPT solver $B(1^\lambda, y_0, \mathcal{D}(p, q_0, \rho))$ for the (ρ, η, γ)-EF-AGCD problem as follows: B starts the simulation of the game SUF-CPA$_{\Pi,A}$ giving y_0 as an evaluation key to A. For an encryption query $(\tau, m) \in \{0,1\}^\lambda \times \mathbb{Z}_Q$ of A, B replies A with $c := (xQ + m) \bmod y_0$, where $x \leftarrow \mathcal{D}(p, q_0, \rho)$. Eventually, A outputs a forgery attempt $((f, \tau_1, \cdots, \tau_l), \hat{c})$. Let I be the set of $i \in \{1, \ldots, l\}$ where τ_i is used, and for each $i = 1, \ldots, l$, choose $c_i \xleftarrow{\$} \mathbb{Z}_{y_0}$ if $i \notin I$, and let c_i be the unique ciphertext returned by the encryption query involving τ_i if $i \in I$. Now B computes $\tilde{c} = f(c_1, \cdots, c_l) \bmod y_0$, and outputs $y_0 / \gcd(y_0, \tilde{c} - \hat{c})$.

For the similar reason as in Theorem 6, the simulation of the encryption oracle by B is exact.

Consider the forgery attempt $((f, \tau_1, \cdots, \tau_l), \hat{c})$ made by A. If it is a type 1 strong forgery, then $\tilde{e} := \mathsf{Eval}(f, (c_i)_{i\in I})$ is not constant. Since y_0 is chosen as a 2^{λ^2}-rough random integer, with negligible exception, y_0 is square-free and $\tilde{e}$ is not constantly $\hat{c}$ modulo a prime factor $p' \geq 2^{\lambda^2}$ of y_0. So, using Schwartz-Zippel lemma,

$$\begin{aligned}
\Pr\big[\tilde{e}(c_j)_{j\notin I} \equiv \hat{c} \bmod y_0 \;|(c_j)_{j\notin I} \xleftarrow{\$} (\mathbb{Z}_{y_0})^{l-|I|}\big]& \\
\leq \Pr\big[\,\tilde{e}(c_j)_{j\notin I} \equiv \hat{c} \bmod p' \mid (c_j)_{j\notin I} \xleftarrow{\$} (\mathbb{Z}_{p'})^{l-|I|}\big]& \\
\leq \frac{\deg f}{p'} \leq \frac{\bar{d}}{2^{\lambda^2}} = \mathit{negl}(\lambda) \enspace .&
\end{aligned}$$

This means that $\tilde{c} = \tilde{e}(c_j)_{j\notin I} \not\equiv \hat{c} \bmod y_0$ with overwhelming probability. If $((f, \tau_1, \cdots, \tau_l), \hat{c})$ is a type 2 strong forgery, then again we have $\tilde{e}(c_j)_{j\notin I} = \tilde{c} \not\equiv \hat{c} \bmod y_0$.

Hence in both cases, we have $\tilde{c} \not\equiv \hat{c} \bmod y_0$, but also $\tilde{c} \equiv \hat{c} \bmod q_0$, since any strong forgery is valid. Therefore, $\gcd(y_0, \hat{c} - \tilde{c}) = q_0$ and the output of B is exactly p with overwhelming probability if the forgery attempt of A is a successful strong forgery. Since A succeeds with non-negligible probability, B outputs the correct answer p with non-negligible probability.

Acknowledgments. We thank the anonymous reviewers for their careful reading of our paper and helpful comments.

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2011-0025127) and was also supported by the year of 2010 Research Fund of the UNIST (Ulsan National Institute of Science and Technology).

References

1. Bellare, M., Desai, A., Jokipii, E., Rogaway, P.: A concrete security treatment of symmetric encryption. In: Proceedings of the 38th Annual Symposium on Foundations of Computer Science. pp. 394–403 (1997)
2. Bellare, M., Goldreich, O., Mityagin, A.: The power of verification queries in message authentication and authenticated encryption. Cryptology ePrint Archive, Report 2004/309 (2004)
3. Bellare, M., Hofheinz, D., Kiltz, E.: Subtleties in the definition of IND-CCA: When and how should challenge decryption be disallowed? J. Cryptol. (2013)
4. Bellare, M., Namprempre, C.: Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. J. Cryptol. 21(4), 469–491 (2008)
5. Boneh, D., Freeman, D.M.: Homomorphic signatures for polynomial functions. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 149–168. Springer, Heidelberg (2011)
6. Brakerski, Z.: Fully homomorphic encryption without modulus switching from classical GapSVP. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 868–886. Springer, Heidelberg (2012)
7. Brakerski, Z., Gentry, C., Halevi, S.: Packed ciphertexts in LWE-based homomorphic encryption. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 1–13. Springer, Heidelberg (2013)
8. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (Leveled) fully homomorphic encryption without bootstrapping. In: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference. pp. 309–325. ITCS 2012, ACM (2012)
9. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. In: Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on. pp. 97–106. IEEE Computer Society CPS (2011)
10. Brakerski, Z., Vaikuntanathan, V.: Fully homomorphic encryption from Ring-LWE and security for key dependent messages. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 505–524. Springer, Heidelberg (2011)
11. Catalano, D., Fiore, D.: Practical homomorphic MACs for arithmetic circuits. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 336–352. Springer, Heidelberg (2013)
12. Catalano, D., Fiore, D., Warinschi, B.: Homomorphic signatures with efficient verification for polynomial functions. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8616, pp. 371–389. Springer, Heidelberg (2014)
13. Cheon, J.H., Coron, J.S., Kim, J., Lee, M.S., Lepoint, T., Tibouchi, M., Yun, A.: Batch fully homomorphic encryption over the integers. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 315–335. Springer, Heidelberg (2013)
14. Coron, J.S., Lepoint, T., Tibouchi, M.: Scale-invariant fully homomorphic encryption over the integers. In: Krawczyk, H. (ed.) PKC 2014. LNCS, vol. 8383, pp. 311–328. Springer, Heidelberg (2014)
15. Coron, J.S., Mandal, A., Naccache, D., Tibouchi, M.: Fully homomorphic encryption over the integers with shorter public keys. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 487–504. Springer, Heidelberg (2011)
16. Coron, J.S., Naccache, D., Tibouchi, M.: Public key compression and modulus switching for fully homomorphic encryption over the integers. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 446–464. Springer, Heidelberg (2012)

17. van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully homomorphic encryption over the integers. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 24–43. Springer, Heidelberg (2010)
18. Dworkin, M.: Recommendation for block cipher modes of operation: Galois/Counter Mode (GCM) for confidentiality and authentication. Special Publication 800-38D, NIST (2007)
19. Gennaro, R., Wichs, D.: Fully homomorphic message authenticators. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013. LNCS, vol. 8270, pp. 301–320. Springer, Heidelberg (2013)
20. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proceedings of the 41st annual ACM symposium on Theory of computing. pp. 169–178. STOC '09, ACM (2009)
21. Gentry, C., Halevi, S.: Implementing Gentry's fully-homomorphic encryption scheme. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 129–148. Springer, Heidelberg (2011)
22. Gentry, C., Halevi, S., Smart, N.P.: Fully homomorphic encryption with polylog overhead. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 465–482. Springer, Heidelberg (2012)
23. Gentry, C., Halevi, S., Smart, N.P.: Homomorphic evaluation of the AES circuit. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 850–867. Springer, Heidelberg (2012)
24. Gorbunov, S., Vaikuntanathan, V.: (Leveled) fully homomorphic signatures from lattices. Cryptology ePrint Archive, Report 2014/463 (2014)
25. Krovetz, T., Rogaway, P.: The software performance of authenticated-encryption modes. In: Joux, A. (ed.) FSE 2011. LNCS, vol. 6733, pp. 306–327. Springer, Heidelberg (2011)
26. Rogaway, P., Bellare, M., Black, J.: OCB: A block-cipher mode of operation for efficient authenticated encryption. ACM Trans. Inf. Syst. Secur. 6(3), 365–403 (2003)
27. Smart, N.P., Vercauteren, F.: Fully homomorphic encryption with relatively small key and ciphertext sizes. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 420–443. Springer, Heidelberg (2010)
28. Wichs, D.: Leveled fully homomorphic signatures from standard lattices. Cryptology ePrint Archive, Report 2014/451 (2014)