Group Homomorphic Encryption
Characterizations, Impossibility Results, and Applications

Frederik Armknecht[1], Stefan Katzenbeisser[2], and Andreas Peter[3]

[1] Universität Mannheim, Germany
armknecht@uni-mannheim.de
[2] Technische Universität Darmstadt, Germany
skatzenbeisser@acm.org
[3] Technische Universität Darmstadt, Germany
andreas.peter@cantab.net

Abstract. We give a complete characterization both in terms of security and design of *all currently existing* group homomorphic encryption schemes, i.e., existing encryption schemes with a group homomorphic decryption function such as ElGamal and Paillier. To this end, we formalize and identify the basic underlying structure of all existing schemes and say that such schemes are of *shift-type*. Then, we construct an abstract scheme that represents all shift-type schemes (i.e., every scheme occurs as an instantiation of the abstract scheme) and prove its IND-CCA1 (resp. IND-CPA) security equivalent to the hardness of an abstract problem called *Splitting Oracle-Assisted Subgroup Membership Problem*, SOAP (resp. *Subgroup Membership Problem*, SMP). Roughly, SOAP asks for solving an SMP instance, i.e., for deciding whether a given ciphertext is an encryption of the neutral element of the ciphertext group, while allowing access to a certain oracle beforehand. Our results allow for contributing to a variety of open problems such as the IND-CCA1 security of Paillier's scheme, or the use of linear codes in group homomorphic encryption.
Furthermore, we design a new cryptosystem which provides features that are unique up to now: Its IND-CPA security is based on the k-linear problem introduced by Shacham, and Hofheinz and Kiltz, while its IND-CCA1 security is based on a *new* k-problem that we prove to have the same progressive property, namely that if the k-instance is easy in the generic group model, the $(k+1)$-instance is still hard.

Keywords: Foundations, Homomorphic Encryption, Public-Key Cryptography, IND-CCA1 Security, Subgroup Membership Problem, k-Linear Problem

1 Introduction

1.1 Motivation

Homomorphic encryption schemes support computation on encrypted data. Such schemes are of particular interest for various applications, such as Outsourcing of Computation [19], Electronic Voting [3, 10, 12, 13], Private Information Retrieval [33], Oblivious Polynomial Evaluation [39], or Multiparty Computation [11].

The most prominent homomorphic encryption schemes, e.g., ElGamal [18], Paillier [42], Damgård-Jurik [16], are homomorphic with respect to a single algebraic operation. That is, the plaintext space forms a group $(G, \circ)$ and, given encryptions of $m, m' \in G$, one can efficiently and securely compute an encryption of $m \circ m'$ without revealing m and m'. We will call such schemes *group homomorphic* encryption schemes. Although fully homomorphic schemes [9, 49, 20, 21, 47], i.e., schemes that allow one to evaluate any circuit over encrypted data without being able to decrypt, provide a much higher flexibility compared to group

homomorphic schemes, the investigation of the latter still represents an important research topic:

1. The majority of existing homomorphic schemes are group homomorphic and there are still many open questions regarding these schemes.
2. For practical applications there is currently no alternative to such schemes.[4]
3. Many constructions of schemes that support more than a single algebraic operation are in particular group homomorphic as well (e.g., [1, 5]).
4. A comprehensive understanding of group homomorphic schemes leads to a better understanding of schemes that are homomorphic in a more general sense, since the underlying structures are very similar.

Over the last decades, a variety of different approaches (and according hardness assumptions and proofs of security) has been investigated for constructing group homomorphic schemes, such as the Quadratic Residuosity Problem [26], the Higher Residuosity Problem [3], the Decisional Diffie-Hellman Problem [18, 44], and the Decisional Composite Residuosity Class Problem [42, 16]. All these schemes have been investigated separately, resulting in the fact that some of them are better understood than others. In particular, much effort has been devoted to proving existing homomorphic schemes IND-CCA1 secure (being the highest possible security level for a homomorphic scheme). For example, since the introduction of Damgård's ElGamal [15] in 1991, many works addressed the problem of characterizing its IND-CCA1 security [25, 50]. Similarly, while an IND-CPA security characterization of ElGamal was given in 1998 (see [48]), the quest for a characterization of its IND-CCA1 security has been in the focus for many years. Only in 2010, the quest concerning these two schemes has finally found an end due to [36]. Finding similar characterizations for remaining homomorphic schemes, e.g., Paillier's scheme, is still an open problem.

1.2 Contribution

In this work, we present a unified view both in terms of security and design on *all currently existing* group homomorphic encryption schemes[5]. On the one hand, this helps to access the kind of challenges mentioned above more easily (and in fact, to answer open questions) and on the other hand provides a systematic procedure for designing new schemes based on given problems. Our concrete contributions are as follows:

Abstract Security Characterization First, we identify and formailze the underlying structure of *all existing* group homomorphic encryption schemes and say that group homomorphic schemes with this structure are of *shift-type*. This particular structure allows us to construct an abstract scheme that represents all shift-type group homomorphic encryption schemes and prove its IND-CCA1 security equivalent to the hardness of a new abstract problem, called the *Splitting Oracle-Assisted Subgroup Membership Problem* (SOAP), meaning that every scheme occurs as an instantiation of the abstract scheme being IND-CCA1

[4] For example, the most efficient implementation [22] of [21] states that the largest variant (for which a security level similar to RSA-1024 is assumed) has a public key of 2.4 GB size and requires about 30 minutes to complete certain operations.

[5] A precise definition will be given in Section 2.2.

secure if and only if the according instantiation of SOAP is hard. This abstract scheme is similar to other existing abstract schemes [17, 21, 23] but is necessarily more general in order to be a representative of *all* shift-type group homomorphic schemes. For a *proper* subclass of shift-type homomorphic schemes, a proof that if an abstract *Subgroup Membership Problem* (SMP) is hard, then the scheme is IND-CPA secure was given in [23]. Our result applies to a *larger* class of homomorphic schemes, namely to *all* shift-type schemes, considers a *higher* security level (IND-CCA1 instead of IND-CPA) and shows IND-CCA1 security *equivalent* to the hardness of SOAP. In fact, a characterization of IND-CPA security through SMP is an immediate byproduct of our results.

Concrete Security Characterization Our abstract security characterizations can be applied to concrete homomorphic schemes by looking at the according instantiations. For example, several results such as the IND-CPA security of ElGamal [48], the IND-CCA1 security of Damgård's ElGamal [15, 25, 36, 50] and the recently proved IND-CCA1 security of ElGamal [36] can be easily derived from our characterizations. Additionally, we use the IND-CCA1 characterization to approach the long standing open question, whether Paillier's homomorphic encryption scheme [42] is IND-CCA1 secure. Clearly, similar concrete security characterizations can be given for all other group homomorphic schemes that are of shift-type.

Furthermore, we derive two impossibility results. First, we show that no group homomorphic scheme with a prime ordered ciphertext group can be IND-CPA secure. Second, we prove that under certain conditions an IND-CPA group homomorphic scheme where the ciphertexts form a linear subspace of $\mathbb{F}^n$ for some prime field $\mathbb{F}$, can never be of shift-type. This partly answers an open question whether using linear codes as ciphertext spaces yield more efficient constructions (see [21]) in the sense that the construction cannot be of shift-type.

Systematic Design Approach Another utilization of our results is a systematic approach for constructing provably secure group homomorphic schemes. By using our abstract scheme and a concrete instantiation of SOAP resp. SMP, one can directly specify a homomorphic scheme that is IND-CCA1 resp. IND-CPA secure if and only if the respective problem is hard.

As an example, we consider the *k-linear problem* [29, 45] which is an alternative to DDH in groups where DDH is easy, e.g., in bilinear groups [30]. After its introduction, many works addressed the problem of constructing cryptographic protocols whose security is based on the k-linear problem (e.g., [4, 27, 29, 31, 35, 40, 45]). Continuing this line of research, we present the first homomorphic scheme that is based on the k-linear problem for $k > 2$ ($k = 1$ is ElGamal [18], $k = 2$ is Linear Encryption [4]). In addition, we introduce a *new* k-problem (an instantiation of SOAP) that we prove to be hard in the generic group model and to have the same progressive property as the k-linear problem. This result might be of independent interest as it can be used to construct new cryptographic protocols with unique features. For instance, we give the first homomorphic scheme that can be instantiated with groups where DDH is easy (e.g., bilinear groups) and is nevertheless provably secure in terms of IND-CCA1.

1.3 Separation from Other Related Work

Aside from the related work that we have already mentioned in the previous sections, there is a substantial number of papers on the construction of IND-CPA (respectively, IND-CCA1,

IND-CCA2) secure encryption schemes. In this regard, we would particularly like to mention the work by Cramer and Shoup [14] who give a generic construction of IND-CPA (respectively, IND-CCA1, IND-CCA2) secure encryption schemes through smooth (respectively, 1-universal, 2-universal) hash proof systems. Furthermore, Peikert and Waters [43] introduce the notion of Lossy Trapdoor Functions (LTFs) and give a generic construction of IND-CCA1 secure encryption schemes from such functions, while Hemenway and Ostrovsky [28] give a generic construction of IND-CCA1 secure group homomorphic encryption schemes through homomorphic hash proof systems, which are known to be constructable, e.g., from the Quadratic Residuosity Problem, the Decisional Diffie-Hellman Problem or the Decisional Composite Residuosity Problem. A somewhat different approach to the construction of IND-CCA1 secure group homomorphic encryption was presented by Prabhakaran and Rosulek [44]. Therein, they build group homomorphic encryption schemes that are secure in an even stronger sense than just being IND-CCA1, namely "homomorphic-CCA" secure.

All these works have in common that they build IND-CCA1 secure schemes from non-interactive assumptions, while we show the IND-CCA1 security equivalent to the hardness of SOAP which then naturally has to be an interactive problem, as IND-CCA1 is. Therefore, we stress that we give *characterizations* of the security of group homomorphic schemes. For all the above mentioned schemes this means that the underlying non-interactive assumption either implies SOAP, or is equivalent to it. In the former case, breaking the underlying assumption would not necessarily break the security of the scheme in question as it is actually equivalent to SOAP which might still be a hard problem. We do not give a generic construction of IND-CCA1 secure group homomorphic schemes from *non-interactive* assumptions. Concerning IND-CPA security on the other hand, this is a different story, as we propose an abstract scheme that encompasses *all* shift-type group homomorphic encryption schemes and hence is a also a generic way to construct IND-CPA secure group homomorphic schemes from non-interactive assumptions. The latter is due to the fact that the corresponding SMP instance is always non-interactive.

1.4 Outline

Throughout the paper, we use standard notation and definitions that are summarized in Section 2. Therein, we also formally define the class of group homomorphic encryption schemes, and recall standard security notions for such schemes. In Section 3, we introduce the notion of shift-type group homomorphic encryption, construct an abstract scheme and prove that it represents all shift-type group homomorphic schemes. We define certain subgroup problems (e.g., SOAP and SMP) in Section 4 and use them to prove the desired security characterizations. Next, we instantiate these problems to analyze the security of existing schemes in Section 5, to show certain impossibility results in Section 6, and to design a new scheme in Section 7.

2 Preliminaries

2.1 General Definitions and Notation

We write $x \longleftarrow X$ if X is a random variable or distribution and x is to be chosen randomly from X according to its distribution. In the case where X is solely a set, $x \xleftarrow{U} X$ denotes

that x is chosen uniformly at random from X. For an algorithm $\mathcal{A}$ we write $x \longleftarrow \mathcal{A}(y)$ if $\mathcal{A}$ outputs x on fixed input y according to $\mathcal{A}$'s distribution. If $\mathcal{A}$ has access to an oracle $\mathcal{O}$, we write $\mathcal{A}^{\mathcal{O}}$. Sometimes, we need to specify the randomness of a probabilistic algorithm $\mathcal{A}$ explicitly. To this end, we interpret $\mathcal{A}$ as a deterministic algorithm $\mathcal{A}(y, r)$, which has access to values r that are chosen *uniformly* at random from some randomness space. Furthermore, if X and Y are random variables taking values in a finite set S, we define the *statistical difference* between X and Y as $\mathrm{Dist}(X, Y) := \frac{1}{2} \cdot \sum_{s \in S} |\Pr[X = s] - \Pr[Y = s]|$. If $\mathrm{Dist}(X, Y) \leq \epsilon$, we say that X and Y are *ϵ-close*.

For a group $\mathcal{G}$, we denote the neutral element by 1, and denote the binary operation on $\mathcal{G}$ by "$\cdot$", i.e., $\mathcal{G}$ is written in *multiplicative notation*. We recall that a subgroup $\mathcal{N}$ of a group $\mathcal{G}$ is said to be *normal* if $z \cdot n \cdot z^{-1} \in \mathcal{N}$ for all $z \in \mathcal{G}, n \in \mathcal{N}$. In particular, this means that if $\mathcal{G}$ is an abelian group, then every subgroup $\mathcal{N}$ is normal. For a finite (not necessarily abelian) group $\mathcal{G}$, a non-trivial, proper normal subgroup $\mathcal{N}$ of $\mathcal{G}$, and a fixed system of representatives $\mathcal{R} \subseteq \mathcal{G}$ of $\mathcal{G}/\mathcal{N}$, we recall the following fact:

Fact 1 *Let τ be the restriction to $\mathcal{R}$ of the canonical surjection $\mathcal{G} \to \mathcal{G}/\mathcal{N}, z \mapsto z \cdot \mathcal{N}$. Now since $\mathcal{R}$ is a system of representatives of $\mathcal{G}/\mathcal{N}$, every $z \in \mathcal{G}$ can be uniquely written as $z = r \cdot n$ with $r \in \mathcal{R}$ and $n \in \mathcal{N}$. Therefore, τ is a bijection and there is a group structure on $\mathcal{R}$ that is inherited from $\mathcal{G}/\mathcal{N}$: For $r, r' \in \mathcal{R}$, we define $r \odot r' := \tau^{-1}(\tau(r) \cdot \tau(r'))$. We denote the element in $\mathcal{R}$ that corresponds to the neutral element in $\mathcal{G}/\mathcal{N}$ by $\mathbf{1}$. It is easy to verify that with the defined operation $\odot$, $\mathcal{R}$ becomes a group with neutral element $\mathbf{1}$. In addition, we know that $\mathcal{R} \cap \mathcal{N} = \mathbf{1}$, since $\mathcal{R} \subseteq \mathcal{G}$ is a system of representatives of $\mathcal{G}/\mathcal{N}$.*

If $f : X \to Y$ is a mapping between two sets X and Y, we write $\mathrm{dom}(f) = X$ for the *domain* of f and $\mathrm{im}(f)$ for its *image*. In addition, we write $f|_S$ for the *restriction* of f to a subset $S \subseteq X$, i.e. $f|_S : S \to Y$ with $f|_S(s) := f(s)$ for all $s \in S$. If X and Y are groups (multiplicatively written), and f is a group homomorphism, we write $\ker(f) := \{x \in X \mid f(x) = 1\}$ for the *kernel* of f. If f is surjective, we write $f^{-1}(y) := \{x \in X \mid f(x) = y\}$ for the *preimage* of y under f for $y \in Y$. Surjective group homomorphisms are also called *group epimorphisms*.

We describe computational problems P through experiments $\mathbf{Exp}^{\mathrm{P}}_{\mathcal{A},G}(\lambda)$ for given probabilistic algorithms $\mathcal{A}$ and G that run in time polynomial in a given parameter λ. The output of $\mathbf{Exp}^{\mathrm{P}}_{\mathcal{A},G}(\lambda)$ is always defined to be a single bit. We then say that *problem P is hard (relative to G)* if for all probabilistic polynomial time (PPT) algorithms $\mathcal{A}$ there exists a negligible function $\mathtt{negl}$ such that

$$\left| \Pr[\mathbf{Exp}^{\mathrm{P}}_{\mathcal{A},G}(\lambda) = 1] - \frac{1}{2} \right| \leq \mathtt{negl}(\lambda).$$

2.2 Group Homomorphic Public Key Encryption

The central notion in this paper is that of *group homomorphic encryption*. Basically, a public key encryption scheme is called *group homomorphic*, if its decryption algorithm is a group homomorphism. Since there are some subtleties to take care of, the following definition gives a precise formalization of this notion.

Definition 1 (Group Homomorphic Encryption). *A public key encryption scheme $\mathcal{E} = (G, E, D)$ is called* group homomorphic*, if for every output (pk, sk) of $G(\lambda)$, the plaintext space $\mathcal{P}$ and the ciphertext space $\widehat{\mathcal{C}}$ are (written in multiplicative notation) non-trivial groups such that*

- *the* set of all encryptions $\mathcal{C} := \{c \in \widehat{\mathcal{C}} \mid c \longleftarrow E_{pk}(m), m \in \mathcal{P}\}$ *is a non-trivial subgroup of* $\widehat{\mathcal{C}}$
- *the restricted decryption* $D^*_{sk} := D_{sk}|_{\mathcal{C}}$ *is a group epimorphism, i.e.*

$$D^*_{sk} \text{ is surjective and } \forall c, c' \in \mathcal{C} : D_{sk}(c \cdot c') = D_{sk}(c) \cdot D_{sk}(c')$$

- *sk contains an efficient* decision function $\delta : \widehat{\mathcal{C}} \to \{0, 1\}$ *such that*

$$\delta(c) = 1 \iff c \in \mathcal{C}$$

- *the decryption on* $\widehat{\mathcal{C}} \setminus \mathcal{C}$ *returns the symbol* $\perp$.

Remark 1. All "classical" homomorphic encryption schemes [15, 16, 18, 23, 24, 26, 38, 41, 42] are indeed group homomorphic in terms of Definition 1. We note that for almost all these schemes, we have $\widehat{\mathcal{C}} = \mathcal{C}$ which lets the decision function be trivial. In these cases, the decryption function is a group epimorphism on the whole of $\widehat{\mathcal{C}}$ and the special symbol $\perp$ is not needed. In fact, we only introduced the decision function to encompass Damgård's ElGamal [15].

Remark 2. Furthermore, we note that it is straightforward to extend all of our results in this paper to *ring* homomorphic encryption schemes, which are defined in precisely the same way as group homomorphic schemes, except that every occurrence of the notion "group" is replaced by "ring" in Definition 1.

We show that the set of encryptions of $1 \in \mathcal{P}$ has a certain group-theoretic structure. For this, we define

$$\mathcal{C}_m := \{c \in \mathcal{C} \mid D_{sk}(c) = m\}$$

as the *set of all encryptions of* $m \in \mathcal{P}$.

Lemma 1. *Let* $\mathcal{E} = (G, E, D)$ *be a group homomorphic encryption scheme that does not necessarily have a decision function* δ. *Then,*

1. $\mathcal{C}_m = E_{pk}(m, r) \cdot \mathcal{C}_1$ *for all* $m \in \mathcal{P}$ *and all random* r. *It follows that the set* $\{E_{pk}(m, r) \mid m \in \mathcal{P}\}$ *for a fixed* r *is a system of representatives of* $\mathcal{C}/\mathcal{C}_1$
2. $\mathcal{C}_1$ *is a proper* normal *subgroup of* $\mathcal{C}$ *such that* $|\mathcal{C}_1| = |\mathcal{C}_m|$ *for all* $m \in \mathcal{P}$.

Proof. We fix a random r and $m \in \mathcal{P}$. Let $c \in \mathcal{C}_m$ and set $c_1 := c \cdot E_{pk}(m, r)^{-1}$. Then, $D_{sk}(c_1) = m \cdot m^{-1} = 1$, i.e. $c_1 \in \mathcal{C}_1$. Therefore, $c = E_{pk}(m, r) \cdot c_1 \in E_{pk}(m, r) \cdot \mathcal{C}_1$. Conversely, let $c_1 \in \mathcal{C}_1$. Then, $D_{sk}(E_{pk}(m, r) \cdot c_1) = m \cdot 1 = m$, i.e. $E_{pk}(m, r) \cdot c_1 \in \mathcal{C}_m$. The first statement of the lemma follows immediately.

With respect to the second claim, we show by contradiction that $\mathcal{C}_1 \neq \mathcal{C}$. Therefore, assume that $\mathcal{C}_1 = \mathcal{C}$. Since the decryption D^*_{sk} is surjective, this means that $\mathcal{P}$ is a trivial group, which contradicts the definition of a homomorphic scheme. Now, by looking at the definition of $\mathcal{C}_1$, we see that $\mathcal{C}_1 = \ker(D^*_{sk})$. Therefore, $\mathcal{C}_1$ is a *normal* subgroup of $\mathcal{C}$ (e.g., [34, p. 13]). The last claim is an immediate consequence of the equality $\mathcal{C}_m = E_{pk}(m, r) \cdot \mathcal{C}_1$.

□

2.3 Security Notions for Public Key Encryption Schemes

We briefly recall the three security notions *indistinguishability under chosen-plaintext attack* (IND-CPA), *indistinguishability under (non-adaptive) chosen-ciphertext attack* (IND-CCA1) and *indistinguishability under adaptive chosen-ciphertext attack* (IND-CCA2) for public key encryption schemes (cf. [2, Definition 2.1]) and explain their role in the group homomorphic case.

Let $\mathcal{E} = (G, E, D)$ be a public key encryption scheme. We will write $\mathcal{O}_i(\cdot) = \varepsilon$, where $i \in \{1, 2\}$, for an oracle function that always returns the empty string ε on any input. For $\text{atk} \in \{\text{cpa}, \text{cca1}, \text{cca2}\}$, a given algorithm $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and parameter λ, we consider the following experiment:

Experiment $\mathbf{Exp}^{\text{ind-atk}}_{\mathcal{A},G}(\lambda)$:

1. $(pk, sk) \longleftarrow G(\lambda)$
2. $(m_0, m_1, s) \longleftarrow \mathcal{A}_1^{\mathcal{O}_1(\cdot)}(pk)$ where $m_0, m_1 \in \mathcal{P}$ and s a state of $\mathcal{A}_1$
3. Choose $b \xleftarrow{U} \{0, 1\}$ and compute $c \longleftarrow E_{pk}(m_b)$
4. $d \longleftarrow \mathcal{A}_2^{\mathcal{O}_2(\cdot)}(m_0, m_1, s, c)$ where $d \in \{0, 1\}$
5. The output of the experiment is defined to be 1 if $d = b$ and 0 otherwise,

$$\begin{array}{rl} & \text{if atk} = \text{cpa} \ \text{ then } \mathcal{O}_1(\cdot) = \varepsilon \qquad \text{and } \mathcal{O}_2(\cdot) = \varepsilon \\ \text{where} & \text{if atk} = \text{cca1 then } \mathcal{O}_1(\cdot) = D_{sk}(\cdot) \text{ and } \mathcal{O}_2(\cdot) = \varepsilon \\ & \text{if atk} = \text{cca2 then } \mathcal{O}_1(\cdot) = D_{sk}(\cdot) \text{ and } \mathcal{O}_2(\cdot) = D_{sk}(\cdot). \end{array}$$

If atk = cca2, we further require that $\mathcal{A}_2$ is not allowed to ask its oracle to decrypt the challenge ciphertext c.

We say that $\mathcal{E}$ is IND-ATK *secure (relative to G)* if the advantage

$$\left| \Pr[\mathbf{Exp}^{\text{ind-atk}}_{\mathcal{A},G}(\lambda) = 1] - \frac{1}{2} \right| \text{ is negligible for all PPT algorithms } \mathcal{A},$$

where ATK $\in$ {CPA, CCA1, CCA2}. Bellare et al. [2] show that IND-CCA2 is strictly stronger than IND-CCA1, which in turn is strictly stronger than IND-CPA.

For reasons of completeness, we prove the following well-known result.

Theorem 1 (No IND-CCA2 Security). *Any group homomorphic encryption scheme $\mathcal{E} = (G, E, D)$, that does not necessarily have a decision function δ, is insecure in terms of* IND-CCA2.

Proof. On input the public key pk, the adversary $\mathcal{A}_1$ outputs two non-zero randomly chosen plaintexts $m_0, m_1 \in \mathcal{P}$ with $m_0 \neq m_1$. The challenger chooses a random bit $b \in \{0, 1\}$ and computes the challenge ciphertext $c \longleftarrow E_{pk}(m_b)$. Upon receiving the challenge, $\mathcal{A}_2$ computes $c_i \longleftarrow (c \cdot E_{pk}(m_i)^{-1})$ for $i \in \{0, 1\}$, and asks the decryption oracle for the decryptions of c_0 and c_1. By definition, one of these decryptions is 1, and $\mathcal{A}_2$ outputs the index $d \in \{0, 1\}$ of the decryption that corresponds to 1. Therefore, the advantage of $\mathcal{A}$ in the IND-CCA2 game is $\frac{1}{2}$, which is non-negligible. □

Due to this Theorem, we know that IND-CCA1 is the strongest of the three security notions for group homomorphic encryption schemes.

We remark that there exist three additional, standard security notions: *Non-malleability* with respect to CPA, CCA1 and CCA2. For details on these, we refer to [2] and note that, for obvious reasons, no group homomorphic encryption scheme can be secure in terms of these notions. Therefore, we do not consider these non-malleability notions. Also, we note that non-standard variants, e.g., [7, 44], lie outside of the scope of this paper.

3 Shift-Type Group Homomorphic Encryption

When looking at all the currently existing group homomorphic encryption schemes (see Section 5 for examples), one notices a certain structure in the encryption procedure that all these schemes have in common. Roughly speaking, the encryption procedure takes a plaintext and adds some "noise" – this noise happens to be an encryption of 1. Formally, this intuition is captured in the following definition.

Definition 2. *A group homomorphic encryption scheme $\mathcal{E} = (G, E, D)$ is said to be of* shift-type*, if the encryption algorithm satisfies the following equation for all random r and all plaintexts $m \in \mathcal{P}$:*

$$E_{pk}(m, r) = E_{pk}(m, \rho) \cdot E_{pk}(1, r),$$

where ρ is a public value from the randomness space such that $E_{pk}(1, \rho) = 1$.

This definition allows us to define an abstract scheme that we prove to be shift-type group homomorphic. Additionally, we show that this abstract scheme encompasses all shift-type group homomorphic schemes and thereby *all existing* group homomorphic schemes. We note that in previous works, similar abstract schemes have been defined [17, 21, 23]. However, none of the previous schemes is general enough to encompass *all existing* group homomorphic schemes. Therefore, we introduce our new scheme, which we call GIFT (Generic shIFt-Type) due to its generality in terms of Definition 2.

Definition 3 (GIFT scheme). GIFT *is a public key encryption scheme $\mathcal{E}_G = (G, E, D)$ with*

Key Generation: *G takes a security parameter λ as input and outputs a tuple (pk, sk) where pk is the public key that contains descriptions of*

- *a non-trivial group $\mathcal{P}$ of plaintexts and a non-trivial group $\widehat{\mathcal{C}}$ of ciphertexts together with a non-trivial subgroup $\mathcal{C} \leq \widehat{\mathcal{C}}$ that will act as the set of encryptions*
- *a non-trivial, proper normal subgroup $\mathcal{N}$ of $\mathcal{C}$ such that $|\mathcal{C}/\mathcal{N}| = |\mathcal{P}|$*
- *an efficient isomorphism $\varphi : \mathcal{P} \longrightarrow \mathcal{R}$ where $\mathcal{R} \subseteq \mathcal{C}$ (not necessarily a subgroup but certainly a group, cf. Remark 1) is a system of representatives of $\mathcal{C}/\mathcal{N}$,*

and sk is the secret key that contains

- *an efficient description of $\varphi^{-1} \circ \nu$ with the epimorphism $\nu : \mathcal{C} \to \mathcal{R}$ such that $\nu(c)$ is the unique representative $r \in \mathcal{R}$ with $c = r \cdot n$ for some $n \in \mathcal{N}$.*
- *an efficient function $\delta : \widehat{\mathcal{C}} \to \{0, 1\}$ such that $\delta(c) = 1 \iff c \in \mathcal{C}$.*

Encryption: *E takes the public key pk and a message $m \in \mathcal{P}$ as input and outputs the ciphertext $c := \varphi(m) \cdot n \in \mathcal{C}$ where $n \longleftarrow \mathcal{N}$.*

Decryption: *D takes the secret key sk and a ciphertext $c \in \widehat{\mathcal{C}}$ as input. If $\delta(c) = 0$, it outputs $\perp$, otherwise it outputs the plaintext $\varphi^{-1}(\nu(c)) \in \mathcal{P}$.*

Remark 3. In GIFT we know that $\mathbf{1} \in \mathcal{N}$,[6] so

$$\begin{aligned}\mathcal{C}_1 &= \{c \in \mathcal{C} \mid \varphi^{-1}(\nu(c)) = 1\} = \{c \in \mathcal{C} \mid \nu(c) = \mathbf{1}\} \\ &= \{c \in \mathcal{C} \mid \mathbf{1} \cdot c^{-1} \in \mathcal{N}\} = \mathcal{N},\end{aligned}$$

i.e. $\mathcal{N}$ is the group of all encryptions of 1.

Next, we prove that GIFT indeed is a shift-type group homomorphic encryption scheme, and that every such scheme can be described in terms of GIFT.

Theorem 2 (Generality). *Every shift-type group homomorphic encryption scheme can be described in terms of* GIFT*, and vice versa.*

Proof. We start by proving that GIFT $\mathcal{E}_{\mathrm{G}} = (G, E, D)$ fulfills Definition 1. By the definition of $\mathcal{E}_{\mathrm{G}}$, it suffices to show the correctness of the scheme and that D^*_{sk} is a group epimorphism.

The correctness can be readily seen, since we know by definition that $\nu(r) = r$ for all $r \in \mathcal{R}$ which implies $\nu(\varphi(m)) = \varphi(m)$ and $\nu(n) = \mathbf{1}$ for all $m \in \mathcal{P}$ and all $n \in \mathcal{N}$. Using that ν and φ are homomorphisms, this yields for all $m \in \mathcal{P}$:

$$\varphi^{-1}(\nu(\varphi(m) \cdot n)) = \varphi^{-1}(\nu(\varphi(m)) \cdot \nu(\mathbf{1})) = \varphi^{-1}(\varphi(m) \cdot \mathbf{1}) = m.$$

Clearly, $D^*_{sk} = \varphi^{-1} \circ \nu$ is an epimorphism since it is the composition of two epimorphisms with $\mathrm{im}(\nu) = \mathrm{dom}(\varphi^{-1})$. It is trivial to see that $\mathcal{E}_{\mathrm{G}}$ is of shift-type.

Conversely, let $\mathcal{E} = (G, E, D)$ be a shift-type group homomorphic scheme and let (pk, sk) be an output of $G(\lambda)$ (pk includes value ρ). We define $\mathcal{N} := \mathcal{C}_1$, which is a proper normal subgroup of $\mathcal{C}$ by Lemma 1. We consider the algorithm $\varphi(\cdot) := E_{pk}(\cdot, \rho)$ that takes messages $m \in \mathcal{P}$ as input. Then, φ is an isomorphism on $\mathcal{P}$ since its inverse φ^{-1} is given by the epimorphism $D_{sk}|_{\mathcal{R}}$ where $\mathcal{R} := \mathrm{im}(\varphi)$. By Lemma 1, we know that $\mathcal{R}$ is a system of representatives of $\mathcal{C}/\mathcal{N}$. Then, we also know that $|\mathcal{P}| = |\mathcal{R}| = |\mathcal{C}/\mathcal{N}|$. Next, we define a PPT algorithm $\overline{E}$ that takes the same inputs as E, i.e., the public key pk and a message $m \in \mathcal{P}$ (written deterministically it also takes a random[7] value z as input), and then does the following:

1. Compute $n := E_{pk}(1, z)$.
2. Output $\overline{c} := \varphi(m) \cdot n$.

We show that $\overline{E_{pk}}$ is an encryption algorithm as required in GIFT:

1. By definition, we have $n \in \mathcal{N} = \mathcal{C}_1$, meaning that we use $E_{pk}(1, \cdot)$ as the sampling algorithm for $\mathcal{N}$.
2. The output $\overline{c}$ of $\overline{E_{pk}}(m)$ has the form $\varphi(m) \cdot n$ with $n \in \mathcal{N}$, as required.

Since $\mathcal{E}$ is of shift-type, we know that $\overline{E}_{pk}$ and E_{pk} have the same output.

All remaining compontents of GIFT are given as follows: By considering $\nu : \mathcal{C} \to \mathcal{R}$ as $\nu := \varphi \circ D_{sk}|_{\mathcal{C}}$, one easily sees that $D_{sk}(c) = \varphi^{-1}(\nu(c))$, if $c \in \mathcal{C}$. Otherwise, i.e. if $\delta(c) = 0$, we have $D_{sk}(c) = \perp$. Hence, we have successfully described $\mathcal{E}$ in terms of GIFT. □

This description of all shift-type group homomorphic schemes allows us to restrict our attention to GIFT. We will make use of this fact in the next sections.

[6] Recall that we denoted the representative in $\mathcal{R}$ of $1 \cdot \mathcal{N}$ by $\mathbf{1}$.

[7] Recall that we interpret PPT algorithms as deterministic algorithms by given them an additional input z that is chosen *uniformly* at random from some randomness space (cf. Section 2).

4 On the Security of Group Homomorphic Encryption Schemes

4.1 Subgroup Problems

In [23], Gjøsteen introduces a computational problem, called *Splitting Problem*, together with a related decisional problem, called *Subgroup Membership Problem*. We recall these two problems and start with the former. For our results on the characterization of group homomorphic schemes in Section 4.2, we need to extend Gjøsteen's definition of the Splitting Problem, as we will explain momentarily.

Let $\widehat{\mathcal{G}}$ be a finite (not necessarily abelian) group, $\mathcal{G}$ a non-trivial subgroup of $\widehat{\mathcal{G}}$, $\mathcal{N}$ a non-trivial, proper normal subgroup of $\mathcal{G}$, and $\mathcal{R} \subseteq \mathcal{G}$ a fixed system of representatives of $\mathcal{G}/\mathcal{N}$. Furthermore, we let $\delta : \widehat{\mathcal{G}} \to \{0,1\}$ with $\delta(z) = 1 \iff z \in \mathcal{G}$ be an efficient decision function.[8]

We recall that every $z \in \mathcal{G}$ can be uniquely written as $z = r \cdot n$ with $r \in \mathcal{R}$ and $n \in \mathcal{N}$ and that there is a natural group structure on $\mathcal{R}$ that is inherited from $\mathcal{G}/\mathcal{N}$ (cf. Remark 1). Moreover, we notice that the following map is a bijection:

$$\mathcal{R} \times \mathcal{N} \to \mathcal{G} \text{ given by } (r,n) \mapsto r \cdot n.$$

We denote its *inverse* by σ and call σ the *splitting map* for $(\mathcal{G},\mathcal{N},\mathcal{R})$.

Informally, the Splitting Problem SP for $(\mathcal{G},\mathcal{N},\mathcal{R})$ is to compute $\sigma(z)$ for a randomly given $z \in \mathcal{G}$. Before we give a formal definition of SP, we note that our definition extends Gjøsteen's in that it considers a system of representatives that need not be a subgroup of $\mathcal{G}$, while Gjøsteen always assumes it to be a subgroup. In addition, we allow $\mathcal{G}$ to be a non-abelian group, while Gjøsteen only considers the abelian case. Now let G be a PPT algorithm that takes a security parameter λ as input and outputs $(\mathcal{G},\mathcal{N},\mathcal{R})$ where $\mathcal{G},\mathcal{N}$ and $\mathcal{R}$ are descriptions of the respective groups defined above. Consider the following experiment for given algorithms G, $\mathcal{A}$ and parameter λ:

Experiment $\mathbf{Exp}^{\mathsf{SP}}_{\mathcal{A},G}(\lambda)$:

1. $(\mathcal{G},\mathcal{N},\mathcal{R}) \longleftarrow G(\lambda)$
2. $(r,n) \longleftarrow \mathcal{A}(\mathcal{G},\mathcal{N},\mathcal{R},z)$ where $r \in \mathcal{R}, n \in \mathcal{N}$ and $z \xleftarrow{U} \mathcal{G}$
3. The output of the experiment is defined to be 1 if $z = r \cdot n$ and 0 otherwise.

This experiment defines the *Splitting Problem* SP *(relative to G)*.

Next, we recall the Subgroup Membership Problem. Let G be a PPT algorithm that takes a security parameter λ as input and outputs descriptions $(\mathcal{G},\mathcal{N})$ of a non-trivial, proper subgroup $\mathcal{N}$ of a (not necessarily abelian) finite group $\mathcal{G}$. Consider the following experiment for a given algorithm G, algorithm $\mathcal{A}$ and parameter λ:

Experiment $\mathbf{Exp}^{\mathsf{SMP}}_{\mathcal{A},G}(\lambda)$:

1. $(\mathcal{G},\mathcal{N}) \longleftarrow G(\lambda)$
2. Choose $b \xleftarrow{U} \{0,1\}$. If $b = 1$: $z \longleftarrow \mathcal{G}$. Otherwise: $z \longleftarrow \mathcal{N}$.
3. $d \longleftarrow \mathcal{A}(\mathcal{G},\mathcal{N},z)$ where $d \in \{0,1\}$

[8] In the following two definitions, we do neither need the decision function nor the group $\widehat{\mathcal{G}}$. The importance of these two objects will become clear later when we define the new problem SOAP.

4. The output of the experiment is defined to be 1 if $d = b$ and 0 otherwise.

This experiment defines the *Subgroup Membership Problem* SMP *(relative to G)* which, informally, states that given $(\mathcal{G}, \mathcal{N}, z)$ where $z \longleftarrow \mathcal{G}$, one has to decide whether $z \in \mathcal{N}$ or not.

It is easy to see that if one can efficiently solve the Splitting Problem for $(\mathcal{G}, \mathcal{N}, \mathcal{R})$ one can also solve the Subgroup Membership Problem for $(\mathcal{G}, \mathcal{N})$: Let $z \in \mathcal{G}$ be the challenge of the SMP for $(\mathcal{G}, \mathcal{N})$. By using the SP solver, we can compute $\sigma(z) = (r, n)$ and we have the relation that $z \in \mathcal{N}$ *if and only if* $r = \mathbf{1}$. So deciding whether $z \in \mathcal{N}$ amounts to deciding whether $r = \mathbf{1}$ which is easy since the neutral element $\mathbf{1}$ of $\mathcal{R}$ is always included in the description of $\mathcal{R}$ (cf. Section 2).

To mention just one of the many concrete instantiations of these two problems, we note that the Computational Diffie-Hellman Problem is an instance of the Splitting Problem, while the corresponding Decisional Diffie-Hellman Problem is an instance of the Subgroup Membership Problem. Further details and other famous examples can be found in Section 5. Also, we want to mention that some other interesting complexity-theoretic results on the SMP can be found in [23, Section 2.1].

At this point, we are in a position that allows us to define a new abstract problem of which two very special cases occur in [36]. Therein, it is proven that the hardness of one of these problems is equivalent to the IND-CCA1 security of ElGamal, while the other's is equivalent to that of Damgård's ElGamal. Informally, the new problem that we will call the *Splitting Oracle-Assisted Subgroup Membership Problem* (SOAP) is situated in the same setting as the Splitting Problem (recall the groups $\widehat{\mathcal{G}}, \mathcal{G}, \mathcal{N}, \mathcal{R}$ and the decision function δ) and consists of two phases. In the first phase the adversary is given access to an oracle $\mathcal{O}_{\mathsf{SP}}^{\widehat{\mathcal{G}},\mathcal{G},\mathcal{N},\mathcal{R},\delta}(\cdot)$ that either solves the Splitting Problem for $(\mathcal{G}, \mathcal{N}, \mathcal{R})$ or outputs the special symbol $\perp$ if the input was not an element of $\mathcal{G}$. In the second/challenge phase, the adversary has to solve the Subgroup Membership Problem for $(\mathcal{G}, \mathcal{N})$. Before we define this problem formally, we remark that it will allow us to deduce characterizations of IND-CCA1 security of all group homomorphic encryption schemes in Section 4.2. In particular, the characterizations for ElGamal and Damgård's ElGamal [36] immediately derive from our generic results.

We let G be a PPT algorithm that takes a security parameter λ as input and outputs descriptions $(\widehat{\mathcal{G}}, \mathcal{G}, \mathcal{N}, \mathcal{R}, \delta)$ of a non-trivial, proper normal subgroup $\mathcal{N}$ of a group $\mathcal{G}$ that is itself a subgroup of a finite group $\widehat{\mathcal{G}}$, a system of representatives $\mathcal{R} \subseteq \mathcal{G}$ of $\mathcal{G}/\mathcal{N}$, and a decision function $\delta : \widehat{\mathcal{G}} \to \{0, 1\}$ given by $\delta(z) = 1 \iff z \in \mathcal{G}$. We consider the following experiment for a given algorithm G, algorithm $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and parameter λ:

Experiment $\mathbf{Exp}_{\mathcal{A},G}^{\mathsf{SOAP}}(\lambda)$:

1. $(\widehat{\mathcal{G}}, \mathcal{G}, \mathcal{N}, \mathcal{R}, \delta) \longleftarrow G(\lambda)$
2. $s \longleftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{SP}}^{\widehat{\mathcal{G}},\mathcal{G},\mathcal{N},\mathcal{R},\delta}(\cdot)}(\widehat{\mathcal{G}}, \mathcal{G}, \mathcal{N}, \mathcal{R}, \delta)$ where s is a state of $\mathcal{A}_1$
3. Choose $b \xleftarrow{U} \{0, 1\}$. If $b = 1$: $z \longleftarrow \mathcal{G}$. Otherwise: $z \longleftarrow \mathcal{N}$
4. $d \longleftarrow \mathcal{A}_2(\widehat{\mathcal{G}}, \mathcal{G}, \mathcal{N}, \mathcal{R}, \delta, s, z)$ where $d \in \{0, 1\}$
5. The output of the experiment is defined to be 1 if $d = b$ and 0 otherwise.

This experiment defines the *Splitting Oracle-Assisted Subgroup Membership Problem (relative to G)*, denoted by SOAP. We note that the splitting oracle $\mathcal{O}_{\mathsf{SP}}^{\widehat{\mathcal{G}},\mathcal{G},\mathcal{N},\mathcal{R},\delta}(\cdot)$ does not solve a

random instance of SP, rather it solves the Splitting Problem for $(\mathcal{G}, \mathcal{N}, \mathcal{R})$ which are the parameters of the corresponding SMP the adversary has to solve in the challenge phase. Therefore, we say that the splitting oracle solves the *static* Splitting Problem (SSP), while "static" in this context refers to the SMP instance the adversary has to solve in the SOAP game. This is why we sometimes denote SOAP by $\mathsf{SMP}^{\mathsf{SSP}}$ following the notation of [36].

Examples of concrete instantiations of all just described subgroup problems can be found in Section 5.1. In particular, we refer to Section 7, where we introduce new instantiations of these problems which we use to construct new group homomorphic schemes with interesting properties.

4.2 Security Characterization

Our aim is to characterize all shift-type group homomorphic encryption schemes in terms of the three standard security notions IND-CPA, IND-CCA1 and IND-CCA2 for public key encryption schemes (cf. Section 2.3). Recall that by Theorem 1, we know that for group homomorphic encryption schemes IND-CCA1 is the strongest of the three security notions. Therefore, characterizing shift-type group homomorphic schemes in terms of this notion is highly desirable.

Theorem 3 (Characterization of IND-CCA1 Security). *Let $\mathcal{E} = (G, E, D)$ be a shift-type group homomorphic encryption scheme. Then:*

$$\mathcal{E} \textit{ is } \mathsf{IND\text{-}CCA1} \textit{ secure (relative to } G) \iff \mathsf{SOAP} \textit{ is hard (relative to } G).$$

Proof. "⇐": By Theorem 2, we know that we can restrict our attention to the GIFT scheme. Therefore, we think of $\mathcal{E}$ being a particular instance of GIFT and assume that $\mathcal{E}$ is not IND-CCA1 secure, i.e. there exists a PPT algorithm $\mathcal{A}^{\mathrm{cca1}} = (\mathcal{A}_1^{\mathrm{cca1}}, \mathcal{A}_2^{\mathrm{cca1}})$ that breaks the security with non-negligible advantage $f(\lambda)$. We derive a contradiction by constructing a PPT algorithm $\mathcal{A}^{\mathrm{soap}} = (\mathcal{A}_1^{\mathrm{soap}}, \mathcal{A}_2^{\mathrm{soap}})$ that successfully solves SOAP with advantage $\frac{1}{2}f(\lambda)$.

Since SOAP and IND-CCA1 are both considered relative to G, $\mathcal{A}_1^{\mathrm{soap}}$ can simply forward the public key $pk = (\mathcal{P}, \widehat{\mathcal{C}}, \mathcal{C}, \mathcal{N}, \mathcal{R}, \varphi)$ of the output of $G(\lambda)$ to $\mathcal{A}_1^{\mathrm{cca1}}$. If $\mathcal{A}_1^{\mathrm{cca1}}$ queries the decryption oracle for a decryption of some ciphertext $c \in \widehat{\mathcal{C}}$, $\mathcal{A}_1^{\mathrm{soap}}$ asks the oracle $\mathcal{O}_{\mathsf{SP}}^{\widehat{\mathcal{C}},\mathcal{C},\mathcal{N},\mathcal{R},\delta}(c)$ on input c which outputs the element $\sigma(c) = (r, n) \in \mathcal{R} \times \mathcal{N}$ if $\delta(c) = 1$ and $\perp$ otherwise. In the former case, it is readily seen that $r = \nu(c)$ and so $\mathcal{A}_1^{\mathrm{soap}}$ forwards the correct plaintext $\varphi^{-1}(r)$ to $\mathcal{A}_1^{\mathrm{cca1}}$ (recall that we consider GIFT). In the latter case, $\mathcal{A}_1^{\mathrm{soap}}$ simply forwards $\perp$ to $\mathcal{A}_1^{\mathrm{cca1}}$.

After the query phase of $\mathcal{A}_1^{\mathrm{cca1}}$ is over, $\mathcal{A}_1^{\mathrm{cca1}}$ outputs two messages $m_0, m_1 \in \mathcal{P}$ to $\mathcal{A}_2^{\mathrm{soap}}$. The SOAP challenger chooses a bit $b \xleftarrow{U} \{0,1\}$ and sends the challenge $c \in \mathcal{C}$ to $\mathcal{A}^{\mathrm{soap}}$, who then chooses a bit $d \xleftarrow{U} \{0,1\}$ and sends the challenge $c_d := E_{pk}(m_d) \cdot c$ to $\mathcal{A}_2^{\mathrm{cca1}}$. Now, $\mathcal{A}_2^{\mathrm{cca1}}$ outputs a bit d' and sends it back to $\mathcal{A}_2^{\mathrm{soap}}$ which sends $b' := d \oplus d'$ to the SOAP challenger.

We have the following relations: If $b = 0$, then $c \in \mathcal{C}_1$ and c_d is a correct encryption of the message m_d. Hence, $\mathcal{A}_2^{\mathrm{cca1}}$ makes the right guess with advantage $f(\lambda)$, i.e. $\Pr[b' = b | b = 0] \geq \frac{1}{2} + f(\lambda)$. If $b = 1$, then $c \in \mathcal{C}$ and c_d looks like a random encryption. Hence, $\mathcal{A}_2^{\mathrm{cca1}}$ guesses d

with no advantage, i.e. $\Pr[b' = b|b = 1] = \frac{1}{2}$. We have shown:

$$\Pr[\mathbf{Exp}^{\mathsf{SOAP}}_{\mathcal{A}^{\mathrm{soap}},G}(\lambda) = 1] = \sum_{\beta\in\{0,1\}} \Pr[b' = b|b = \beta] \cdot \Pr[b = \beta]$$
$$\geq \frac{1}{2} \cdot \left(\frac{1}{2} + f(\lambda) + \frac{1}{2}\right) = \frac{1}{2} + \frac{1}{2}f(\lambda).$$

"⇒": For the converse, we assume that there is a PPT algorithm $\mathcal{A}^{\mathrm{soap}} = (\mathcal{A}^{\mathrm{soap}}_1, \mathcal{A}^{\mathrm{soap}}_2)$ that solves SOAP with advantage $f(\lambda)$. Similarly to what we have done above, we construct a PPT algorithm $\mathcal{A}^{\mathrm{cca1}} = (\mathcal{A}^{\mathrm{cca1}}_1, \mathcal{A}^{\mathrm{cca1}}_2)$ that successfully breaks the IND-CCA1 security with advantage $f(\lambda)$.

Similarly to the above, $\mathcal{A}^{\mathrm{cca1}}_1$ forwards the part $(\widehat{\mathcal{C}}, \mathcal{C}, \mathcal{N}, \mathcal{R}, \delta)$ of the output of $G(\lambda)$ to $\mathcal{A}^{\mathrm{soap}}_1$. If $\mathcal{A}^{\mathrm{soap}}_1$ queries the oracle $\mathcal{O}^{\widehat{\mathcal{C}},\mathcal{C},\mathcal{N},\mathcal{R},\delta}_{\mathsf{SP}}(c)$ on input $c \in \widehat{\mathcal{C}}$, $\mathcal{A}^{\mathrm{cca1}}_1$ asks the decryption oracle for a decryption of c that outputs the plaintext $m := D_{sk}(c) = \varphi^{-1}(\nu(c))$ if $\delta(c) = 1$ and $\perp$ otherwise. In the former case, we notice that $\varphi(m) \in \mathcal{R}$ and so $\mathcal{A}^{\mathrm{cca1}}_1$ sends the correct Splitting Problem solution $(\varphi(m), \varphi(m) \cdot c^{-1})$ to $\mathcal{A}^{\mathrm{soap}}_1$. In the latter case, $\mathcal{A}^{\mathrm{cca1}}_1$ simply forwards $\perp$ to $\mathcal{A}^{\mathrm{soap}}_1$. After the query phase of $\mathcal{A}^{\mathrm{soap}}_1$ is over, $\mathcal{A}^{\mathrm{cca1}}_1$ outputs two messages $m_0, m_1 \in \mathcal{P}$. The IND-CCA1 challenger chooses a bit $b \xleftarrow{U} \{0,1\}$ and sends the challenge $c_b \longleftarrow E_{pk}(m_b)$ to $\mathcal{A}^{\mathrm{cca1}}_2$, who then computes $c := c_b \cdot E_{pk}(m_0)^{-1} \in \mathcal{C}$ and sends the challenge c to $\mathcal{A}^{\mathrm{soap}}_2$. Now, $\mathcal{A}^{\mathrm{soap}}_2$ returns a bit d' to $\mathcal{A}^{\mathrm{cca1}}_2$ that then outputs $b' := d'$ to the IND-CCA1 challenger.

We have the following relations: If $b = 0$, then $c \in \mathcal{C}_1$ and $\mathcal{A}^{\mathrm{soap}}_2$ guesses b with advantage $f(\lambda)$, i.e. $\Pr[b' = b|b = 0] \geq \frac{1}{2} + f(\lambda)$. If $b = 1$, then $c \in \mathcal{C} \setminus \mathcal{C}_1$ and $\mathcal{A}^{\mathrm{soap}}_2$ guesses b again with advantage $f(\lambda)$, i.e. $\Pr[b' = b|b = 1] \geq \frac{1}{2} + f(\lambda)$. Therefore, we have shown:

$$\Pr[\mathbf{Exp}^{\text{ind-cca1}}_{\mathcal{A}^{\mathrm{cca1}},G}(\lambda) = 1] = \sum_{\beta\in\{0,1\}} \Pr[b' = b|b = \beta] \cdot \Pr[b = \beta]$$
$$\geq \frac{1}{2} \cdot (1 + 2f(\lambda)) = \frac{1}{2} + f(\lambda).$$

□

A careful study of the proof of Theorem 3 shows that, as a special case, we have also proven a characterization of IND-CPA security. It is interesting to see that for this characterization the decision function δ is *not* needed.

Theorem 4 (Characterization of IND-CPA Security). *Let $\mathcal{E} = (G, E, D)$ be a shift-type group homomorphic encryption scheme that does not necessarily have a decision function δ. Then:*

$$\mathcal{E} \textit{ is } \mathsf{IND\text{-}CPA} \textit{ secure (relative to } G) \iff \mathsf{SMP} \textit{ is hard (relative to } G).$$

Proof. If $\mathcal{A}^{\mathrm{cpa}} = (\mathcal{A}^{\mathrm{cpa}}_1, \mathcal{A}^{\mathrm{cpa}}_2)$ is a successful adversary on IND-CPA with advantage $f(\lambda)$, then the adversary $\mathcal{A}^{\mathrm{soap}}_2$ from the first part of the proof of Theorem 3 successfully solves SMP with advantage $\frac{1}{2}f(\lambda)$ when changing every occurrence of $\mathcal{A}^{\mathrm{cca1}}$ by $\mathcal{A}^{\mathrm{cpa}}$ in the proof.

Conversely, let $\mathcal{A}^{\mathrm{smp}}$ be a successful adversary on SMP with advantage $f(\lambda)$. We consider the adversary $\mathcal{A}^{\mathrm{cca1}} = (\mathcal{A}^{\mathrm{cca1}}_1, \mathcal{A}^{\mathrm{cca1}}_2)$ from the second part of the proof of Theorem 3. Since here, $\mathcal{A}^{\mathrm{cca1}}_1$ has no oracle access, it outputs two random messages $m_0, m_1 \in \mathcal{P}$ with $m_0 \neq m_1$. Then, following the proof of Theorem 3 while changing every occurrence of $\mathcal{A}^{\mathrm{soap}}$ by $\mathcal{A}^{\mathrm{smp}}$ in the proof, $\mathcal{A}^{\mathrm{cca1}}$ successfully solves IND-CPA with advantage $f(\lambda)$. □

We note that in [23], Gjøsteen already proved one of the implications for a much *smaller* class of group homomorphic schemes, namely that if SMP is hard, then $\mathcal{E}$ is IND-CPA secure. We stress that our result is more powerful since we consider the larger class of shift-type schemes (that encompasses all existing group homomorphic schemes) and since we give the first proof of the other implication which is the key ingredient for the highly desirable characterization. Interestingly enough, compared to the IND-CCA1 case, the IND-CPA characterization also holds for shift-type group homomorphic schemes that do not have a decision function δ.

5 Security Characterization of Existing Schemes

One application of our approach is an easy characterization of IND-CPA and IND-CCA1 security of existing schemes. For example, the results on the IND-CPA resp. IND-CCA1 security of ElGamal, given in [48] resp. [36], and for Damgård's ElGamal, given in [15] resp. [36], are direct consequences as the next section shows. More interesting is the application to open problems, and as an example, we will consider the IND-CCA1 security of Paillier's homomorphic encryption scheme [42] in Section 5.2.

5.1 Known Security Characterizations

We want to give two concrete instantiations of the three subgroup problems that we have defined in Section 4.1, and instantiations of GIFT. Furthermore, we look at two schemes whose security is based on the respective problem instantiation, namely ElGamal [18] and Damgård's ElGamal [15]. Finally, we analyse their security through our characterization results, Theorems 3 and 4. Interestingly enough, the well-known security proofs of these schemes [36, 48] immediately derive from our general results. For other famous examples of instantiations, we refer to [23] and [24], while we refer to Sections 5.2 and 7 of this paper for *new* instantiations.

ElGamal. In GIFT, we let $\widehat{\mathcal{C}} = \mathcal{C} = \mathcal{G} \times \mathcal{G}$ be the direct product of a cyclic group $\mathcal{G}$ (multiplicatively written) of prime order p with generator g. Since $\widehat{\mathcal{C}} = \mathcal{C}$, the decision function $\delta : \widehat{\mathcal{C}} \to \{0,1\}$ is trivial, i.e. always outputs 1. We set $\mathcal{P} := \mathcal{G}$ and let $\mathcal{N} = \langle (g,h) \rangle$ be a subgroup of $\mathcal{C}$ generated by $(g,h) \in \mathcal{C}$ where $h := g^a$ for a secret $a \xleftarrow{U} \mathbb{Z}_p$. Since $\mathcal{N} \cap \mathcal{R} = \{(1,1)\}$ where $\mathcal{R} := \langle (1,g) \rangle \leq \mathcal{C}$ with $|\mathcal{R}| = p$, we know that $\mathcal{R}$ is a system of representatives of $\mathcal{C}/\mathcal{N}$ (the isomorphism is given by $(1,g^r) \mapsto (1,g^r) \cdot \mathcal{N}$). Trivially, we have the efficient isomorphism $\varphi : \mathcal{P} \to \mathcal{R}$ given by $g^r \mapsto (1,g^r)$. Also, we define an efficient epimorphism $\nu : \mathcal{C} \to \mathcal{R}$ given by $(g^r, g^s) \mapsto (1, g^s \cdot g^{-ar})$. We have successfully defined the ingredients of the public key pk and the secret key sk as required in GIFT. Clearly, this instantiation of GIFT is ElGamal [18].

Next, we look at the three subgroup problems for this particular instantiation. First, recall that a triple of elements $(g_1, g_2, g_3) = (g^a, g^b, g^\gamma) \in \mathcal{G}^3$ is called a Diffie-Hellman triple if $\gamma = a \cdot b$. Furthermore, one can easily check that $(g_2, g_3) \in \mathcal{N}$ if and only if (h, g_2, g_3) is a Diffie-Hellman triple. The Splitting Problem for $(\mathcal{C}, \mathcal{N}, \mathcal{R})$ is the computational Diffie-Hellman (CDH) problem for (h, c_1), since the splitting map $\sigma : \mathcal{C} \to \mathcal{R} \times \mathcal{N}$ is given by $(c_1, c_2) \mapsto ((1, c_2 \cdot c_1^{-a}), (c_1, c_1^a))$. The Subgroup Membership Problem for $(\mathcal{C}, \mathcal{N})$ is the decisional Diffie-Hellman (DDH) problem for (h, c_1, c_2), and SOAP for $(\widehat{\mathcal{C}}, \mathcal{C}, \mathcal{N}, \mathcal{R}, \delta)$ is the problem $\mathsf{DDH}^{\mathsf{SCDH}}$ where SCDH denotes the static computational Diffie-Hellman problem (cf. [36]).

In the ElGamal instantiation, we see that Theorem 4 states that ElGamal is IND-CPA secure if and only if DDH is hard, while Theorem 3 states that it is IND-CCA1 secure if and only if $\mathsf{DDH}^{\mathsf{SCDH}}$ is hard. The former characterization was proven in [48], while the latter was proven in [36].

Damgård's ElGamal. Again, we look at a concrete instantiation of GIFT. Here, we let $\widehat{\mathcal{C}} = \mathcal{G}^3$ be the direct product of a prime ordered cyclic group $\mathcal{G}$ with generator g, and set $\mathcal{P} := \mathcal{G}$. Furthermore, we choose random secrets $a, b \xleftarrow{U} \mathbb{Z}_p$, compute the values $h := g^a, s := g^s$ and set $\mathcal{C} := \langle (g,h) \rangle \times \mathcal{G}$. For a ciphertext $c = (c_1, c_2, c_3) \in \widehat{\mathcal{C}}$ we see that $c \in \mathcal{C} \iff c_2 = c_1^a$. Therefore, we have found an efficient decision function $\delta : \widehat{\mathcal{C}} \to \{0,1\}$. Next, we set $\mathcal{N} := \langle (g,h,s) \rangle$ and $\mathcal{R} := \langle (1,1,g) \rangle$. Since $\mathcal{N} \cap \mathcal{R} = \{(1,1,1)\}$ and $|\mathcal{R}| = p$, we see that $\mathcal{R}$ is a system of representatives of $\mathcal{C}/\mathcal{N}$ (the isomorphism is given by $(1,1,g^r) \mapsto (1,1,g^r) \cdot \mathcal{N}$). We immediately derive an efficient isomorphism $\varphi : \mathcal{P} \to \mathcal{R}$ given by $g^r \mapsto (1,1,g^r)$ and define the map $\nu : \mathcal{C} \to \mathcal{R}$ by $(g^r, h^r, g^t) \mapsto (1,1,g^t \cdot g^{-br})$. We have successfully defined the ingredients of the public key pk and the secret key sk as required in GIFT and easily see that this instantiation is Damgård's ElGamal [15].

By considering the Splitting Problem for $(\mathcal{C}, \mathcal{N}, \mathcal{R})$ in this particular instantiation, we see that the splitting map $\sigma : \mathcal{C} \to \mathcal{R} \times \mathcal{N}$ is given by $(c_1, c_2, c_3) \mapsto ((1,1,c_3 \cdot c_1^{-b}), (c_1, c_2, c_1^b))$. Therefore, this Splitting Problem coincides with the CDH problem with parameters (g, s, g^r) for random $r \xleftarrow{U} \mathbb{Z}_p$; In [36], this problem is denoted by CDEG. The Subgroup Membership Problem for $(\mathcal{C}, \mathcal{N})$ is the DDH problem with parameters (g, s, g^r, g^t) for random $r \xleftarrow{U} \mathbb{Z}_p$ and $t \in \mathbb{Z}_p$; In [36], this problem is denoted by DDEG. Finally, SOAP for $(\widehat{\mathcal{C}}, \mathcal{C}, \mathcal{N}, \mathcal{R}, \delta)$ is the problem $\mathsf{DDEG}^{\mathsf{SCDEG}}$ where SCDEG is the static CDEG (cf. [36]).

For this instantiation, i.e. for Damgård's ElGamal, Theorem 4 states that it is IND-CPA secure if and only if DDEG is hard, while Theorem 3 states that it is IND-CCA1 secure if and only if $\mathsf{DDEG}^{\mathsf{SCDEG}}$ is hard. The former characterization was proven in [15], while the latter was very recently proven in [36].

5.2 Paillier's Scheme

We briefly recall Paillier's homomorphic encryption scheme [42] by plugging the appropriate parameters into GIFT. Therefore, let $n = pq$ be an RSA-modulus and set $\widehat{\mathcal{C}} := \mathcal{C} := \mathbb{Z}_{n^2}^*$, $\mathcal{P} := \mathbb{Z}_n$ and $\mathcal{N} := \{r^n \bmod n^2 \mid r \in \mathbb{Z}_n^*\}$. Recall the following homomorphism

$$\mathcal{E}_g : \mathbb{Z}_n \times \mathbb{Z}_n^* \longrightarrow \mathbb{Z}_{n^2}^* \text{ with } \mathcal{E}_g(x,y) := g^x \cdot y^n \bmod n^2$$

for an element $g \in \mathbb{Z}_{n^2}^*$. It is known that $\mathcal{E}_g$ is an isomorphism if $g = 1 + n$ [8] or, more generally, if g is a multiple of n [42]. In these cases, there is a unique tuple $(x,y) \in \mathbb{Z}_n \times \mathbb{Z}_n^*$ for each $\omega \in \mathbb{Z}_{n^2}^*$ with $\mathcal{E}_g(x,y) = \omega$. The value x is called the *n-th residuosity class of ω (with respect to g)*, denoted by $[\![\omega]\!]_g$. The problem of computing $[\![\omega]\!]_g$ for given $\omega \in \mathbb{Z}_{n^2}^*$ and g is called the *Computational Composite Residuosity* (CCR) problem. Paillier showed that when the factorization of n is known, it is easy to compute $[\![\omega]\!]_g$ given ω and g. The problem of deciding whether $x = [\![\omega]\!]_g$, given ω, g and x, is called *Decisional Composite Residuosity* (DCR) problem.

In the following, we fix $g \in \mathbb{Z}_{n^2}^*$ such that $\mathcal{E}_g$ is an isomorphism and consider the subgroup $\mathcal{R} := \langle h \rangle$ of $\mathcal{C}$ generated by $h := 1 + n$. In [14, Section 8.2.1], it is shown that $\mathcal{R} = \{1 +$

$an \bmod n^2 \mid a \in \mathbb{Z}_n\}$ with $|\mathcal{R}| = n = |\mathcal{C}/\mathcal{N}|$ (in particular, we can efficiently solve discrete logarithm in $\mathcal{R}$ due to this simple structure). In fact, $\mathcal{R}$ is a system of representatives of $\mathcal{C}/\mathcal{N}$:

Lemma 2. *Let $\pi : \mathcal{C} \to \mathcal{C}/\mathcal{N}$ be the canonical epimorphism, i.e. $\pi(c) := c \cdot \mathcal{N}$. Then, the map $\rho := \pi|_{\mathcal{R}} : \mathcal{R} \to \mathcal{C}/\mathcal{N}$ is an isomorphism, i.e. $\mathcal{R}$ is a system of representatives of $\mathcal{C}/\mathcal{N}$.*

Proof. Since ρ, as the restriction of π, is a homomorphism and $|\mathcal{R}| = |\mathcal{C}/\mathcal{N}|$, it suffices to show that ρ is injective. Therefore, let $h^a \bmod n^2 \in \ker(\rho) = \mathcal{N} \cap \mathcal{R}$ for some $a \in \mathbb{Z}_n$, i.e. there exists $z \in \mathbb{Z}_n^*$ such that $h^a \equiv z^n \pmod{n^2}$. But $\mathcal{N}$ is a group and so there exists an element $y \in \mathbb{Z}_n^*$ such that $y^n \cdot z^n \equiv 1 \pmod{n^2}$, i.e. $h^a \cdot y^n \equiv 1 \pmod{n^2}$. This in turn implies that $\mathcal{E}_h(a, y) \equiv 1 \pmod{n^2}$. But $\mathcal{E}_h$ is an isomorphism, i.e. $(a, y) = (0, 1) \in \mathbb{Z}_n \times \mathbb{Z}_n^*$ which implies $h^a \bmod n^2 = 1 \bmod n^2$ and so ρ is injective. □

Trivially, we have the isomorphism $\varphi : \mathcal{P} \to \mathcal{R}$ given by $m \mapsto 1 + mn \bmod n^2$. By [42, Lemma 1+Lemma 2], we know that the "class function" $[\![\cdot]\!]_g : \mathbb{Z}_{n^2}^* \to \mathbb{Z}_n$ is a group epimorphism and so the mapping $\nu : \mathcal{C} \to \mathcal{R}$ given by $c \mapsto h^{[\![c]\!]_g \bmod n} \bmod n^2$ is a group epimorphism. It can be efficiently computed when the factorization of n is known [42, Theorem 1]. Since we can solve discrete logarithms in $\mathcal{R}$ very efficiently, computing $\nu(c)$ is equivalent to computing $[\![c]\!]_g$.

We have successfully defined the public key $pk = (n, g)$ and the secret key $sk = (p, q)$ in GIFT. The resulting scheme is Paillier's homomorphic encryption scheme [42]. Observe that the splitting map $\sigma : \mathcal{C} \to \mathcal{R} \times \mathcal{N}$ is given by $\omega \mapsto ([\![\omega]\!]_g, \omega \cdot g^{-[\![\omega]\!]_g})$. We immediately see that the SP in this instantiation is the CCR problem. Furthermore, $\mathcal{N}$ contains by definition all elements $r^n \bmod n^2$ for $r \in \mathbb{Z}_n^*$. Therefore, the SMP for $(\mathcal{C}, \mathcal{N})$ is the DCR problem. As a consequence of Theorems 3 and 4, we get the following characterizations of the security of Paillier's scheme:

Theorem 5 (Security Characterization of Paillier). *Paillier's scheme is* IND-CCA1 *(resp.* IND-CPA*) secure if and only if* $\mathsf{DCR}^{\mathsf{SCCR}}$ *(resp. the* DCR *problem) is hard.*

We note that the $\mathsf{DCR}^{\mathsf{SCCR}}$ is a new (though naturally arising) problem and so a thorough analysis of its hardness is advisable. Since such an analysis lies outside of the scope of this paper, we leave it as an open question.

Damgård and Jurik proposed an extension of Paillier's scheme to a generalised group structure [16]. We stress that we can achieve a similiar characterization of the IND-CCA1 security of their scheme by applying similar thoughts as the above.

6 Impossibility Results

In this section, we show two impossibility results. The first is stated in the following easy corollary:

Corollary 1. *Let $\mathcal{E} = (G, E, D)$ be a shift-type group homomorphic encryption scheme that does not necessarily have a decision function δ. If $\mathcal{C}$ is a group of prime order, then $\mathcal{E}$ is insecure in terms of* IND-CPA*.*

Proof. Since $\mathcal{C}$ has prime order, we know that $\mathcal{C}_1$ is trivial, i.e. it is easy to decide membership in $\mathcal{C}_1$. Hence, the scheme cannot be IND-CPA secure by Theorem 4. □

Of course, this result easily extends to the general case: Whenever $\mathcal{C}_1$ is trivial, we just choose 1 as one of the messages in the IND-CPA challenge and can then simply check whether the challenge ciphertext is the single element in $\mathcal{C}_1$ or not.

The second result is motivated by the question whether IND-CPA secure *code-based* group homomorphic schemes exist. For instance, [1] presents a *symmetric* shift-type group homomorphic scheme (that even allows for a limited amount of multiplications) based on linear codes. The immediate question that arises is, whether this scheme works in the public key setting as well. In [20, p. 10], it is asked more generally, whether it is possible to construct a fully homomorphic scheme that is code-based.

Let $\mathbb{F}$ be a prime field. Recall that a *linear code* of length n and rank k is a linear subspace $C \subseteq \mathbb{F}^n$ of the vector space $\mathbb{F}^n$ such that $\dim(C) = k$. Theorem 4 partly answers the question from above, when the ciphertext space $\widehat{\mathcal{C}}$ is a linear code. We need the following two Lemmata:

Lemma 3. *Let $U \subseteq V$ be a non-trivial linear subspace of a $\mathbb{F}$-vector space V with $\dim(U) = k$ and $\dim(V) = n$. Futhermore, we assume that we can sample from U uniformly at random. For all $1 \leq \ell \leq k$, we have: If $(u_1, \ldots, u_\ell) \xleftarrow{U} U^\ell$, then the probability that $u_1, \ldots, u_\ell$ are linearly independent is $\prod_{i=1}^{\ell}(1 - |\mathbb{F}|^{i-k-1})$.*

In particular, if $\ell = k$, the probability that the tuple $(u_1, \ldots, u_k) \xleftarrow{U} U^k$ is linearly independent equals $\prod_{i=1}^{k}(1 - |\mathbb{F}|^{-i})$.

Proof. The proof works by induction on $1 \leq \ell \leq k$. The case $\ell = 1$ is trivial. So let $\ell > 1$ and let $(u_1, \ldots, u_{\ell-1}) \xleftarrow{U} U^{\ell-1}$. By the induction hypothesis, we know that this is a linearly independent tuple with probability $\prod_{i=1}^{\ell-1}(1 - |\mathbb{F}|^{i-k-1})$. Now, since $\dim(U) = k$, U has precisely $|\mathbb{F}|^k$ many elements. On the other hand, there are precisely $|\mathbb{F}|^{\ell-1}$ many vectors in U that are linearly dependent to $(u_1, \ldots, u_{\ell-1})$, so the probability that $u_1, \ldots, u_{\ell-1}, u_\ell$ are linearly dependent, where $u_\ell \xleftarrow{U} U$, is $|\mathbb{F}|^{\ell-1}/|\mathbb{F}|^k = |\mathbb{F}|^{\ell-k-1}$. In total this means that the tuple $(u_1, \ldots, u_\ell)$ is with probability $\prod_{i=1}^{\ell-1}(1 - |\mathbb{F}|^{i-k-1}) \cdot (1 - |\mathbb{F}|^{\ell-k-1}) = \prod_{i=1}^{\ell}(1 - |\mathbb{F}|^{i-k-1})$ linearly independent. If $\ell = k$, this value equals $\prod_{i=1}^{k}(1 - |\mathbb{F}|^{-i})$. □

This Lemma essentially says that when choosing k vectors of U uniformly at random, the probability that these vectors are linearly dependent is negligible in the size of $\mathbb{F}$, i.e. they form a basis of U, except with negligible probability in $|\mathbb{F}|$. By replacing all occurrences of the uniform distribution in the proof by a distribution that is ϵ-close to the uniform distribution, we immediately see the following consequence.

Lemma 4. *Let $U \subseteq V$ be a non-trivial linear subspace of a $\mathbb{F}$-vector space V with $\dim(U) = k$ and $\dim(V) = n$. Furthermore, let $\mathcal{D}$ be a distribution on U that is ϵ-close to the uniform distribution. If ϵ is negligible in $|\mathbb{F}|$, then the probability that the tuple $(u_1, \ldots, u_k) \longleftarrow U^k$ (sampled according to $\mathcal{D}$) is linearly dependent is negligible in $|\mathbb{F}|$.*

This yields the desired impossibility result:

Theorem 6. *Let $\mathcal{E} = (G, E, D)$ be a shift-type group homomorphic encryption scheme, that does not necessarily have a decision function δ, such that the set of encryptions $\mathcal{C}$ is a k-dimensional linear subspace of $\mathbb{F}^n$ and such that the output distribution of the encryption algorithm is ϵ-close to the uniform distribution for some ϵ that is negligible in $|\mathbb{F}|$. Then, $\mathcal{E}$ is insecure in terms of* IND-CPA *(relative to G).*

In particular this holds if $\mathcal{C}$ (or the ciphertext space $\widehat{\mathcal{C}}$)[9] is a linear code.

Proof. According to Theorem 4, we only have to show that SMP is not hard (relative to G). Therefore, we show that, when given a ciphertext $c \in \mathcal{C}$, there is an efficient algorithm that can decide whether $c \in \mathcal{C}_0$ or not.

By using E_{pk} with input 0, we can efficiently sample from $\mathcal{C}_0$. By Lemma 4, this means that we can efficiently construct a basis $(c_1, \ldots, c_s)$ of $\mathcal{C}_0$, where $s := \dim(\mathcal{C}_0)$, by sampling s times at random from $\mathcal{C}_0$. If $(c_1, \ldots, c_s)$ is linearly dependent, which happens with negligible probability, we sample again until we get a linearly independent tuple.

Note that, since $\mathbb{F}$ is a prime field, $\mathcal{C}_0$ is actually an $\mathbb{F}$-subspace of $\mathcal{C}$ (see [32, Theorem 2.1.8(b)]). On the other hand, the basis vectors $c_1, \ldots, c_s$ of $\mathcal{C}_0$ are vectors in $\mathbb{F}^n$. Therefore, when given an arbitrary ciphertext $c \in \mathcal{C}$, we can efficiently compute the rank r of the matrix $(c, c_1, \ldots, c_s)$. If $r = s$, we know that $c \in \mathcal{C}_0$, otherwise $c \notin \mathcal{C}_0$. □

We remark that the same attack also works in the following settings, making the impossibility result more general:

1. If $\mathcal{E}$ is also homomorphic with respect to the scalar multiplication in $V = \mathbb{F}^n$ (i.e. decryption is $\mathbb{F}$-linear), we do not need the restriction that $\mathbb{F}$ is a prime field.
2. Theorem 6 also holds for arbitrary n-dimensional $\mathbb{F}$-vector spaces V, if there is a (publicly known) efficiently computable isomorphism from V to $\mathbb{F}^n$ (the inversion must be efficiently computable as well). We note that this is not always the case, as is seen by considering ElGamal's encryption scheme (see Section 5.1):
   Certainly, the ciphertext group $\mathcal{C} = \mathcal{G} \times \mathcal{G}$ of ElGamal is a 2-dimensional $\mathbb{F}_p$-vector space, where $\mathcal{G}$ is cyclic group of prime order p. In addition, it is easily seen that the group $\mathcal{C}_1$ of all encryptions of 1 is in fact an $\mathbb{F}_p$-subspace of $\mathcal{C}$. So, if there would be a publicly known and efficiently computable isomorphism $F : \mathcal{C} \to \mathbb{F}_p^2$, Theorem 6 would break ElGamal. Fortunately, we can prove that no such isomorphism can exist:

   Claim. If there exists an efficient isomorphism $F : \mathcal{C} \to \mathbb{F}_p^2$, we can efficiently solve discrete logarithms in $\mathcal{G}$ (which is supposed to be hard in the setting of ElGamal).

   Proof. Assume that $F : \mathcal{C} \to \mathbb{F}_p^2$ is an efficiently computable isomorphism. Let $1 \neq g \in \mathcal{G}$ be an arbitrary element of $\mathcal{G}$, i.e., $\mathcal{G} = \langle g \rangle$. Now, for a given $h \in \mathcal{G}$, we can compute $\log_g(h)$ by computing $\log_{F(g,g)}(F(h,h))$. This works since F is $\mathbb{F}_p$-linear (i.e., $F(h,h) = \log_g(h) \cdot F(g,g)$ and so $\log_{F(g,g)}(F(h,h)) = \log_g(h)$) and solving discrete logarithms in the additive group $\mathbb{F}_p^2$ is easy. □

In the situation of [1], Theorem 6 implies that their scheme is, in the public key setting, insecure in terms of IND-CPA.

7 A Homomorphic Scheme based on *k*-Linear

In [30], Joux and Nguyen point out the need for cryptographic protocols whose security is *not* based on DDH by showing that in bilinear groups, the DDH problem is always easy. This issue

[9] $\mathbb{F}$ is a prime field and so the notion of subgroups coincides with the notion of $\mathbb{F}$-subspaces (see [32, Theorem 2.1.8(b)]). Since we assume $\mathcal{C}$ to be a subgroup of $\widehat{\mathcal{C}}$, it follows that if $\widehat{\mathcal{C}}$ is a linear code, then $\mathcal{C}$ is a linear code as well.

has been addressed by Boneh, Boyen and Shacham in [4] by introducing an alternative to the DDH problem called the *decisional linear problem* and describing a homomorphic encryption scheme that is based on this new problem. Independently of each other, Hofheinz and Kiltz [29], and Shacham [45] gave a generalization of the linear problem to the so-called *decisional k-linear problem* (LP_k). They prove that, in the generic group model [46], LP_{k+1} is hard even if LP_k is easy. Following the warning by Joux and Nguyen, they formulate the need for protocols whose security is based on LP_k. We note that LP_1 is the DDH problem, while LP_2 is the decisional linear problem. Since the introduction of the k-linear problem, many protocols have been designed whose security is based on it, e.g. [4, 27, 29, 31, 35, 40, 45] to name just a few. However, a homomorphic encryption scheme whose $\mathsf{IND\text{-}CPA}$ security is based on the LP_k for $k > 2$ is still missing.

In this section, we close this gap and do even more. We first recall the *computational* and the *decisional* k-linear problem (CLP_k, resp. LP_k) and formulate the new problem $\mathsf{LP}_k^{\mathsf{SCLP}_k}$ which is an instance of SOAP defined in Section 4.1, whereas SCLP_k is the *static*-CLP_k, i.e. it is defined with respect to the public parameters of the underlying LP_k problem in $\mathsf{LP}_k^{\mathsf{SCLP}_k}$ (cf. Section 4.1). Trivially, we have the relation that if $\mathsf{LP}_{k+1}^{\mathsf{SCLP}_{k+1}}$ is easy, then so is $\mathsf{LP}_k^{\mathsf{SCLP}_k}$. In addition, it is shown in [36] that $\mathsf{DDH}^{\mathsf{SCDH}} = \mathsf{LP}_1^{\mathsf{SCLP}_1}$ is hard for generic groups which proves that $\mathsf{LP}_k^{\mathsf{SCLP}_k}$ is also hard. Furthermore, we prove in the generic group model that if $\mathsf{LP}_k^{\mathsf{SCLP}_k}$ is easy, then $\mathsf{LP}_{k+1}^{\mathsf{SCLP}_{k+1}}$ is still hard. Thus, we have found a new problem with the same desirable property as LP_k. This result might be of independent interest as it can be used to construct new cryptographic protocols. For instance, we introduce a homomorphic encryption scheme whose $\mathsf{IND\text{-}CCA1}$ security is based on $\mathsf{LP}_k^{\mathsf{SCLP}_k}$ while its $\mathsf{IND\text{-}CPA}$ security is based on the decisional k-linear problem. Thereby giving the first $\mathsf{IND\text{-}CCA1}$ secure homomorphic scheme that can be instantiated with groups where DDH is easy, e.g., bilinear groups.

The k-Linear Problem Fix $k \in \mathbb{N}$. Let $\widehat{\mathcal{C}} := \mathcal{C} := \mathcal{G}^{k+1}$ where $\mathcal{G}$ is a cyclic group of prime order p, generated by g. Furthermore, we choose $a_i \xleftarrow{U} \mathbb{Z}_p^*$ for $i = 1, \ldots, k$ and set $\mathcal{N} := \{(g^{a_1 r_1}, \ldots, g^{a_k r_k}, g^{\sum_{i=1}^k r_i}) \mid \forall i = 1, \ldots, k : r_i \in \mathbb{Z}_p\}$ and $\mathcal{R} := \langle 1 \rangle^k \times \mathcal{G}$. Clearly, $|\mathcal{N}| = p^k$, $|\mathcal{R}| = p$ and $\mathcal{N} \cap \mathcal{R} = \{(1, \ldots, 1)\}$. Therefore, $\mathcal{R}$ is a system of representatives of $\mathcal{C}/\mathcal{N}$ (the isomorphism is given by $(1, \ldots, 1, g^r) \mapsto (1, \ldots, 1, g^r) \cdot \mathcal{N}$). The splitting map $\sigma : \mathcal{C} \to \mathcal{R} \times \mathcal{N}$ for $(\mathcal{C}, \mathcal{N}, \mathcal{R})$ is given by

$$(c_1, \ldots, c_{k+1}) \mapsto \left(\left(1, \ldots, 1, c_{k+1} \cdot \left(\prod_{i=1}^k c_i^{a_i^{-1}} \right)^{-1} \right), \left(c_1, \ldots, c_k, \prod_{i=1}^k c_i^{a_i^{-1}} \right) \right).$$

Now, the CLP_k is the Splitting Problem for $(\mathcal{C}, \mathcal{N}, \mathcal{R})$ while the LP_k is the Subgroup Membership Problem for $(\mathcal{C}, \mathcal{N})$. As a *new* problem, we define $\mathsf{LP}_k^{\mathsf{SCLP}_k}$ as the instance of SOAP for $(\widehat{\mathcal{C}}, \mathcal{C}, \mathcal{N}, \mathcal{R}, \delta)$ where the decision function δ is trivial since $\widehat{\mathcal{C}} = \mathcal{C}$.

The Cryptosystem and Its Security Let $\widehat{\mathcal{C}}$, $\mathcal{C}$, $\mathcal{N}$, $\mathcal{R}$, δ, g and the a_i's be as in the previous section. Furthermore, we set $\mathcal{P} := \mathcal{G}$. We have the isomorphism $\varphi : \mathcal{P} \to \mathcal{R}$ given by $m \mapsto (1, \ldots, 1, m)$ and the epimorphism $\nu : \mathcal{C} \to \mathcal{R}$ given by $(c_1, \ldots, c_{k+1}) \mapsto \left(1, \ldots, 1, c_{k+1} \cdot \prod_{i=1}^k c_i^{-a_i^{-1}}\right)$. We have successfully defined all the ingredients for GIFT for a fixed $k \in \mathbb{N}$. The resulting cryptosystem can be summarized as follows:

Key Generation: *Input.* Security parameter λ. *Output.* $sk = (a_1, \ldots, a_k)$ and $pk = (p, g, g_1 := g^{a_1}, \ldots, g_k := g^{a_k})$ where $a_i \xleftarrow{U} \mathbb{Z}_p^*$ for $i = 1, \ldots, k$ and g is a generator of a cyclic group $\mathcal{G}$ of prime order p such that λ is the length of the binary representation of p.

Encryption: *Input.* Public key pk and plaintext $m \in \mathcal{G}$. *Output.* Ciphertext c with

$$c := (g_1^{r_1}, \ldots, g_k^{r_k}, m \cdot g^{\sum_{i=1}^k r_i}) \text{ where } r_i \xleftarrow{U} \mathbb{Z}_p \text{ for } i = 1, \ldots, k.$$

Decryption: *Input.* Secret key sk and ciphertext $c = (c_1, \ldots, c_{k+1}) \in \mathcal{G}^{k+1}$. *Output.* Plaintext $m := c_{k+1} \cdot \prod_{i=1}^k c_i^{-a_i^{-1}}$.

When instantiated with $k = 1$ the above cryptosystem is ElGamal [18], while for $k = 2$ it is the *linear encryption scheme* introduced in [4]. For the security of the introduced cryptosystem, Theorems 4 and 3 yield:

Corollary 2. *The above cryptosystem is* IND-CPA *secure (resp.* IND-CCA1 *secure) if and only if* LP_k *(resp.* $\mathsf{LP}_k^{\mathsf{SCLP}_k}$*) is hard.*

Concerning the hardness of the new problem $\mathsf{LP}_k^{\mathsf{SCLP}_k}$, we start with a trivial fact:

Theorem 7 (On the Hardness of $\mathsf{LP}_k^{\mathsf{SCLP}_k}$).

1. *If* $\mathsf{LP}_{k+1}^{\mathsf{SCLP}_{k+1}}$ *is easy, then so is* $\mathsf{LP}_k^{\mathsf{SCLP}_k}$.
2. $\mathsf{LP}_1^{\mathsf{SCLP}_1}$ *is hard in the generic group model (see [36]) and so* $\mathsf{LP}_k^{\mathsf{SCLP}_k}$ *is hard in the generic group model (by using 1.)*

Additionally (and this is the more important result), we show the following:

Theorem 8 ($\mathsf{LP}_k^{\mathsf{SCLP}_k}$ in the Generic Group Model). *In the generic group model, we have the following* Progressive Property*:*

$$\textit{If } \mathsf{LP}_k^{\mathsf{SCLP}_k} \textit{ is easy, then } \mathsf{LP}_{k+1}^{\mathsf{SCLP}_{k+1}} \textit{ is still hard.}$$

7.1 Proof of Theorem 8

Let $\mathcal{G}$ be a cyclic group of prime order p. Similarly to Shacham's proof [45] of the progressive property of LP_k, we prove an even stronger result than Theorem 8 by using multilinear maps [6]. We call an efficient map $e_k : \mathcal{G}^k \to \mathcal{G}_T$ *k-multilinear*, if $e_k(z_1^{r_1}, \ldots, z_k^{r_k}) = e_k(z_1, \ldots, z_k)^{\prod_{i=1}^k r_i}$ for all $z_1, \ldots, z_k \in \mathcal{G}$ and $r_1, \ldots, r_k \in \mathbb{Z}_p$.

In what follows, we show that in generic groups featuring a $(k+1)$-multilinear map $\mathsf{LP}_k^{\mathsf{SCLP}_k}$ is easy, but $\mathsf{LP}_{k+1}^{\mathsf{SCLP}_{k+1}}$ is hard. This result implies Theorem 8.

We make extensive use of Shacham's paper [45], starting with a trivial consequence of one of his results. In Lemma B.1 of [45] it is shown that when given a $(k+1)$-multilinear map, there is an efficient algorithm for deciding LP_k. Immediately, this yields:

Corollary 3. *Given a* $(k+1)$*-multilinear map, there is an efficient algorithm for solving* $\mathsf{LP}_k^{\mathsf{SCLP}_k}$.

Next, we give an upper bound on the success probability of an $\mathsf{LP}_k^{\mathsf{SCLP}_k}$-adversary in the presence of a k-multilinear map. We proof this results along the lines of [45] (wherein a similar results is proven for LP_k).

Lemma 5. *If a q-step ($q \geq 2k$) adversary $\mathcal{A}$ solves $\mathsf{LP}_k^{\mathsf{SCLP}_k}$ in the generic group model (featuring a k-multilinear map), then its success probability is at most $\frac{q\cdot(q+2k+4)^2}{2p}$.*

Proof. First, we stress that the *computational* k-linear problems are all equivalent to each other [45], and we can therefore restrict our attention to the problem $\mathsf{LP}_k^{\mathsf{SCDH}}$. Now, let g_0 be a generator of $\mathcal{G}$, and $a_1, \ldots, a_k, y \xleftarrow{U} \mathbb{Z}_p$. We set $g_i := g_0^{a_i}$ for $i \in \{1, \ldots, k\}$ and $g := g_0^y$. Furthermore, let $r_1, \ldots, r_k, s \xleftarrow{U} \mathbb{Z}_p$ and $d \xleftarrow{U} \{0,1\}$, and set $T_d := g_0^{y\sum_{i=1}^k r_i}$ and $T_{1-d} := g_0^s$. The adversary $\mathcal{A}$ is first given access to an SCDH oracle and then receives the opaque representations for the elements

$$g_0, g_0^{a_1}, \ldots, g_0^{a_k}, g_0^y, g_0^{a_1 r_1}, \ldots, g_0^{a_k r_k}, T_0, T_1. \tag{1}$$

Upon reception, $\mathcal{A}$ outputs a bit d' and wins, if $d' = d$.

Let $Q \leq q$ be the number of queries made by the adversary to the SCDH oracle. In the generic group model, the SCDH oracle is equivalent to the multiplication with the element y (cf., [36]). So in the challenge phase, the adversary $\mathcal{A}$ does not only get the opaque representations for the elements in (1), but also representations of $g_0^{y^2}, \ldots, g_0^{y^{Q+1}}$. As usual in the generic group model [37], we have an algorithm $\mathcal{B}$ that, internally, keeps track of elements handled by $\mathcal{A}$ as polynomials in the ring $\mathbb{Z}_p[A_1, \ldots, A_k, Y, R_1, \ldots, R_k, S]$ and, externally, describes these as arbitrary opaque strings in some sufficiently large domain. It maintains these two representations in two lists $\{(F_i, \xi_i)\}$ and $\{(F_{T,i}, \xi_{T,i})\}$ for elements of $\mathcal{G}$ and $\mathcal{G}_T$, respectively. We assume that the domain for external representations is large enough so that, except with negligible probability, $\mathcal{A}$ can only query for elements it previously obtained from $\mathcal{B}$, and $\mathcal{B}$ never outputs the same opaque representation for two different elements.

Now, in the challenge phase, $\mathcal{A}$ is provided with elements that $\mathcal{B}$ internally represents by the following polynomials:

$$g_0 : F = 1,\ g_1 : F = A_1,\ \ldots,\ g_k : F = A_k,\ g : F = Y,\ \ldots,\ g^{y^{Q+1}} : F = Y^{Q+1}$$
$$\text{and } g_1^{r_1} : F = A_1 R_1,\ \ldots,\ g_k^{r_k} : F = A_k R_k,\ T_0 : F = T_0,\ T_1 : F = T_1.$$

On these elements to which $\mathcal{A}$ is given opaque representations, $\mathcal{A}$ can perform the following operations by using $\mathcal{B}$:

- *Group Action:* On input two elements of $\mathcal{G}$, internally represented as F_1 and F_2, $\mathcal{B}$ adds $F' := F_1 + F_2$ to the representation list of $\mathcal{G}$ (if not already there), and outputs with the corresponding external representation. The group action for $\mathcal{G}_T$ is handled analogously.
- *Inversion:* On input an element of $\mathcal{G}$, internally represented as F, $\mathcal{B}$ adds $F' := -F$ to the representation list of $\mathcal{G}$ (if not already there), and outputs with the corresponding external representation. The inversion for $\mathcal{G}_T$ is handled analogously.
- *Multilinear Map:* On input k elements of $\mathcal{G}$ (internally represented as $F_1, \ldots, F_k$) $\mathcal{B}$ adds $F' := \prod_{i=1}^k F_i$ to the representation list of $\mathcal{G}_T$ (if not already there), and outputs with the corresponding external representation.

We see that for all F on the representation list for $\mathcal{G}$, we have $\deg(F) \leq q$, while for all F_T on the representation list for $\mathcal{G}_T$, we have $\deg(F_T) \leq 2k$. After placing the remaining $q - Q$ queries (recall that $\mathcal{A}$ is allowed to make q steps in total) to these operations, it outputs its guess d' for d.

Now, $\mathcal{B}$ chooses $a_1, \ldots, a_k, y, r_1, \ldots, r_k, s \xleftarrow{U} \mathbb{Z}_p$. If we set

$$A_1 := a_1, \ \ldots, \ A_k := a_k, \ Y := y, \ R_1 := r_1, \ \ldots, \ R_k := r_k, \tag{2}$$

$$T_d := y \cdot \textstyle\sum_{i=1}^{k} r_i, \ T_{1-d} := s, \tag{3}$$

the simulation engineered by algorithm $\mathcal{B}$ is consistent with these values unless there are two distinct polynomials F_1 and F_2 on the representation list for $\mathcal{G}$ or two distinct polynomials $F_{T,1}$ and $F_{T,2}$ on the representation list for $\mathcal{G}_T$ that take on the same value under the assignment above. It remains to show that $\mathcal{A}$ cannot construct such a collision independently of the choice of the random values and that the probability that the choice of random values produces a collision is bounded. We recall that $\mathcal{A}$ additionally has the opaque representations of $y^2, \ldots, y^{Q+1}$ due to the SCDH oracle.

Certainly, the probability that there are at least two equal values among $y, y^2, \ldots, y^{Q+1}$ is negligible in p, and since all the random values are independent of each other, except for the value of $T_d = y \cdot \sum_{i=1}^{k} r_i$, the adversary $\mathcal{A}$ must produce a multiple of $Y \cdot \sum_{i=1}^{k} R_i$, say $F = XY \sum_{i=1}^{k} R_i$ for some non-zero X, only by using the terms in (2) and (3). Clearly, any monomial that can be produced from $A_1, \ldots, A_k, Y, \ldots, Y^{Q+1}, A_1R_1, \ldots, A_kR_k, T_d, T_{1-d}$ by using the above described operations is divisible by A_i if it is divisible by R_i for each $i = 1, \ldots, k$. Furthermore, for every i, each monomial in the expansion of XYR_i in $F = XY \sum_{j=1}^{k} R_j$ must be divisible by A_i, hence $A_i \mid X$ (a formal proof of this fact is given in [45]). Therefore, F is divisible by the $k+2$ monomials $A_1, \ldots, A_k, Y$ and R_i for some i. Since $\mathcal{A}$ only knows Y and its powers $Y^2, \ldots, Y^{Q+1}$, and since no term A_aA_b is known to $\mathcal{A}$ for any a, b, forming F would require taking the product of at least $k+1$ of the polynomials available to the adversary. But the multilinear map only allows for forming the product of at most k terms. Thus, $\mathcal{A}$ cannot produce F and is hence unable to cause a collision.

Finally, we give an upper bound for the probability that a random choice of the values $a_1, \ldots, a_k, y, r_1, \ldots, r_k, s$ causes the same value on two distinct polynomials. Since the degrees of the polynomials in the representation list of $\mathcal{G}$ are upper bounded by q, the probability that two such polynomials have the same evaluation for some random values is at most $\frac{q}{p}$ (over the choice of values) (cf., [46, Lemma 1]). Analogously, this probability is at most $\frac{2k}{p}$ for polynomials in the representation list of $\mathcal{G}_T$ since the degrees of these are upper bounded by $2k$. In the challenge phase, the two representation lists consist together of $2k + Q + 4$ values. When the adversary $\mathcal{A}$ does its remaining $q - Q$ queries, the lists contain at most $q + 2k + 4$ values, and the success probability of $\mathcal{A}$ is bounded by

$$\binom{q+2k+4}{2} \frac{q}{p} \leq \frac{q \cdot (q+2k+4)^2}{2p}.$$

In particular, constant success probability requires $q = \Omega(\sqrt[3]{p})$ steps. □

Therefore, we have proven Theorem 8 by taking Corollary 3 and Lemma 5 together.

8 Conclusion

In this work, we gave a unified view on group homomorphic encryption schemes by identifying and abstracting their most fundamental properties (in particular, by identifying the shift-type structure). This view allowed us to give *complete* characterizations both in terms of

design and security of *all existing* group homomorphic schemes. Beside these all-embracing characterizations, we also deduced new theoretical insights on existing schemes and their security (e.g., regarding Paillier's scheme), on subgroup problems (e.g., the identification of the naturally arising problem SOAP) and derived two impossibility results (e.g., regarding the use of linear codes). On the practical side, our unified framework enables us to construct new encryption schemes quite easily, which we emphasize by giving an example based on the k-linear problem that in particular, led to the construction of a new problem with the same desirable progressive property in the generic group model.

References

1. Frederik Armknecht and Ahmad-Reza Sadeghi. A new approach for algebraically homomorphic encryption. Cryptology ePrint Archive, Report 2008/422, 2008. http://eprint.iacr.org/.
2. Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In Hugo Krawczyk, editor, *CRYPTO*, volume 1462 of *Lecture Notes in Computer Science*, pages 26–45. Springer, 1998.
3. J. Benaloh. *Verifiable secret-ballot elections*. PhD thesis, Yale University, New Haven, CT, USA, 1987.
4. Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matthew K. Franklin, editor, *CRYPTO*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55. Springer, 2004.
5. Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-dnf formulas on ciphertexts. In Joe Kilian, editor, *TCC*, volume 3378 of *Lecture Notes in Computer Science*, pages 325–341. Springer, 2005.
6. Dan Boneh and Alice Silverberg. Applications of multilinear forms to cryptography. *Contemporary Mathematics*, 324:71–90, 2002.
7. Ran Canetti, Hugo Krawczyk, and Jesper Buus Nielsen. Relaxing chosen-ciphertext security. In Dan Boneh, editor, *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 565–582. Springer, 2003.
8. Dario Catalano, Rosario Gennaro, Nick Howgrave-Graham, and Phong Q. Nguyen. Paillier's cryptosystem revisited. In *ACM Conference on Computer and Communications Security*, pages 206–214, 2001.
9. Kai-Min Chung, Yael Kalai, and Salil Vadhan. Improved delegation of computation using fully homomorphic encryption. In Tal Rabin, editor, *Advances in Cryptology CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 483–501. Springer Berlin / Heidelberg, 2010.
10. Josh D. Cohen and Michael J. Fischer. A robust and verifiable cryptographically secure election scheme (extended abstract). In *FOCS*, pages 372–382. IEEE, 1985.
11. Ronald Cramer, Ivan Damgård, and Jesper Buus Nielsen. Multiparty computation from threshold homomorphic encryption. In Birgit Pfitzmann, editor, *EUROCRYPT*, volume 2045 of *Lecture Notes in Computer Science*, pages 280–299. Springer, 2001.
12. Ronald Cramer, Matthew K. Franklin, Berry Schoenmakers, and Moti Yung. Multi-autority secret-ballot elections with linear work. In *EUROCRYPT*, pages 72–83, 1996.
13. Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers. A secure and optimally efficient multi-authority election scheme. In *EUROCRYPT*, pages 103–118, 1997.
14. Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars R. Knudsen, editor, *EUROCRYPT*, volume 2332 of *Lecture Notes in Computer Science*, pages 45–64. Springer, 2002.
15. Ivan Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In Joan Feigenbaum, editor, *CRYPTO*, volume 576 of *Lecture Notes in Computer Science*, pages 445–456. Springer, 1991.
16. Ivan Damgård and Mads Jurik. A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In Kwangjo Kim, editor, *Public Key Cryptography*, volume 1992 of *Lecture Notes in Computer Science*, pages 119–136. Springer, 2001.
17. Michael Fellows and Neal Koblitz. Combinatorial cryptosystems galore! In *Finite Fields: Theory, Applications, and Algorithms*, volume 168 of *Contemporary Mathematics*, pages 51–61. American Mathematical Society, 1993.
18. Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *CRYPTO*, pages 10–18, 1984.

19. Rosario Gennaro, Craig Gentry, and Bryan Parno. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In Tal Rabin, editor, *CRYPTO*, volume 6223 of *Lecture Notes in Computer Science*, pages 465–482. Springer, 2010.
20. Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. crypto.stanford.edu/craig.
21. Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *STOC*, pages 169–178. ACM, 2009.
22. Craig Gentry and Shai Halevi. *Implementing Gentry's Fully-Homomorphic Encryption Scheme*, August 2010. https://researcher.ibm.com/researcher/files/us-shaih/fhe-implementation.pdf.
23. Kristian Gjøsteen. Homomorphic cryptosystems based on subgroup membership problems. In Ed Dawson and Serge Vaudenay, editors, *Mycrypt*, volume 3715 of *Lecture Notes in Computer Science*, pages 314–327. Springer, 2005.
24. Kristian Gjøsteen. Symmetric subgroup membership problems. In Serge Vaudenay, editor, *Public Key Cryptography*, volume 3386 of *Lecture Notes in Computer Science*, pages 104–119. Springer, 2005.
25. Kristian Gjøsteen. A new security proof for damgård's elgamal. In David Pointcheval, editor, *CT-RSA*, volume 3860 of *Lecture Notes in Computer Science*, pages 150–158. Springer, 2006.
26. Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.
27. Jens Groth, Rafail Ostrovsky, and Amit Sahai. Non-interactive zaps and new techniques for nizk. In Cynthia Dwork, editor, *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, pages 97–111. Springer, 2006.
28. Brett Hemenway and Rafail Ostrovsky. Lossy trapdoor functions from smooth homomorphic hash proof systems. *Electronic Colloquium on Computational Complexity (ECCC)*, 16:127, 2009.
29. Dennis Hofheinz and Eike Kiltz. Secure hybrid encryption from weakened key encapsulation. In Alfred Menezes, editor, *CRYPTO*, volume 4622 of *Lecture Notes in Computer Science*, pages 553–571. Springer, 2007.
30. Antoine Joux and Kim Nguyen. Separating decision diffie-hellman from computational diffie-hellman in cryptographic groups. *J. Cryptology*, 16(4):239–247, 2003.
31. Eike Kiltz. Chosen-ciphertext security from tag-based encryption. In Shai Halevi and Tal Rabin, editors, *TCC*, volume 3876 of *Lecture Notes in Computer Science*, pages 581–600. Springer, 2006.
32. Hans Kurzweil and Bernd Stellmacher. *The Theory of Finite Groups: An Introduction*. Springer, 2004.
33. Eyal Kushilevitz and Rafail Ostrovsky. Replication is not needed: Single database, computationally-private information retrieval. In *FOCS*, pages 364–373, 1997.
34. Serge Lang. *Algebra*, volume 211 of *Graduate Texts in Mathematics*. Springer-Verlag, 2002.
35. Allison B. Lewko and Brent Waters. Efficient pseudorandom functions from the decisional linear assumption and weaker variants. In Ehab Al-Shaer, Somesh Jha, and Angelos D. Keromytis, editors, *ACM Conference on Computer and Communications Security*, pages 112–120. ACM, 2009.
36. Helger Lipmaa. On the cca1-security of elgamal and damgård's elgamal. In *Proceedings of Inscrypt 2010*. Springer, 2010. http://research.cyber.ee/∼lipmaa/papers/lip10/. To appear.
37. Ueli M. Maurer. Abstract models of computation in cryptography. In Nigel P. Smart, editor, *IMA Int. Conf.*, volume 3796 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2005.
38. David Naccache and Jacques Stern. A new public key cryptosystem based on higher residues. In *ACM Conference on Computer and Communications Security*, pages 59–66, 1998.
39. Moni Naor and Benny Pinkas. Oblivious polynomial evaluation. *SIAM J. Comput.*, 35(5):1254–1281, 2006.
40. Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In Shai Halevi, editor, *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 18–35. Springer, 2009.
41. Tatsuaki Okamoto and Shigenori Uchiyama. A new public-key cryptosystem as secure as factoring. In *EUROCRYPT*, pages 308–318, 1998.
42. Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT*, pages 223–238, 1999.
43. Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In Cynthia Dwork, editor, *STOC*, pages 187–196. ACM, 2008.
44. Manoj Prabhakaran and Mike Rosulek. Homomorphic encryption with cca security. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *ICALP (2)*, volume 5126 of *Lecture Notes in Computer Science*, pages 667–678. Springer, 2008.
45. Hovav Shacham. A cramer-shoup encryption scheme from the linear assumption and from progressively weaker linear variants. Cryptology ePrint Archive, Report 2007/074, 2007. http://eprint.iacr.org/.

46. Victor Shoup. Lower bounds for discrete logarithms and related problems. In *EUROCRYPT*, pages 256–266, 1997.
47. Nigel P. Smart and Frederik Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In Phong Q. Nguyen and David Pointcheval, editors, *Public Key Cryptography*, volume 6056 of *Lecture Notes in Computer Science*, pages 420–443. Springer, 2010.
48. Yiannis Tsiounis and Moti Yung. On the security of elgamal based encryption. In Hideki Imai and Yuliang Zheng, editors, *Public Key Cryptography*, volume 1431 of *Lecture Notes in Computer Science*, pages 117–134. Springer, 1998.
49. Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In Henri Gilbert, editor, *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 24–43. Springer, 2010.
50. J. Wu and D.R. Stinson. On the security of the elgamal encryption scheme and damgards variant. Cryptology ePrint Archive, Report 2008/200, 2008. http://eprint.iacr.org/.