Formal Notions of Security for Verifiable Homomorphic Encryption [⋆]

Jakub Klemsa(✉) and Ivana Trummová

Czech Technical University in Prague, Czech Republic
jakub.klemsa@fel.cvut.cz, trummiva@fit.cvut.cz

Abstract. Homomorphic encryption enables computations with encrypted data, however, in its plain form, it does not guarantee that the computation has been performed honestly. For the Fully Homomorphic Encryption (FHE), a verifiable variant emerged soon after the introduction of FHE itself, for a single-operation homomorphic encryption (HE), particular verifiable variant has been introduced recently, called the VeraGreg Framework. In this paper, we identify a weakness of List Non-Malleability as defined for the VeraGreg framework—an analogy to the classical Non-Malleability—and suggest its improvement which we show not to be extendable any more in certain sense. Next, we suggest a decomposition of the abstract VeraGreg framework, introduce novel notions of security for the resulting components and show some reductions between them and/or their combinations. Finally, we conjecture that VeraGreg achieves the strongest (and desirable) security guarantee if and only if its building blocks achieve certain, much more tangible properties, in a specific case together with an assumption on hardness of particular kind of the famous Shortest Vector Problem for lattices.

Keywords: Verifiable homomorphic encryption · Formal notions of security · Non-malleability

1 Introduction

In 2009, the first publicly known Fully Homomorphic Encryption (FHE) scheme was introduced by Gentry [9]. One year later, an improvement of FHE was published by Gennaro et al. [8] that allows for verification of a computation that has been performed with encrypted values. Since then, many improvements and modifications to FHE emerged: theoretical advances [10,5], implementations [12,16] as well as enhancements of verifiable variants [4,7].

However, to the best of our knowledge, our recent verifiable homomorphic encryption scheme—the VeraGreg *Framework* [13]—was the first of its kind, i.e., a verifiable homomorphic encryption scheme with single operation. Unlike FHE, there exists a plenty of homomorphic encryption schemes with single operation (HE), e.g., plain/unpadded RSA [15] (multiplicative), Goldwasser-Micali

[⋆] This work was supported by the Grant Agency of the CTU in Prague, grant No. SGS19/109/OHK3/2T/13. The full version available at http:// http://.

cryptosystem [11] (the first additive), or Paillier cryptosystem [14] (additive) which was shown by Armknecht et al. [1] to be even IND-CCA1-secure. Neither of these schemes can verify that a computation has been performed honestly.

Our Contributions

Based on a simple observation, we identify a weakness of the notion of List Non-Malleability (LNM; an analogy to the classical Non-Malleability, NM) as defined in [13] and suggest its improvement that takes this obvious vulnerability into account as well. We show that, in certain sense, it is not possible to further strengthen our improved definition.

Following the VeraGreg instantiation structure introduced in [13] (referred to as the VeraGreg *Scheme*), we suggest a decomposition of the abstract VeraGreg framework into smaller building blocks. For these components, we define formal notions of security, in particular modifications of non-malleability. In our theorems, we show some reductions between these notions.

Finally, we conjecture that VeraGreg achieves the strongest security notion if and only if its underlying constructs achieve our novel notions in combination with known results. In a specific case, we strengthen the attacker and postulate a lattice problem to be hard in order to satisfy the strongest security guarantee. As a result, we suggest a simplification to the VeraGreg scheme.

2 Preliminaries

Notations and Symbols

Definition 1 (Multiset). *Let B be a finite set. We call any $\mathcal{B} \in \mathbb{Z}^{|B|} =: \mathcal{M}(B)$ a* multiset *of elements of B, interpreted as a set with (possibly negative) number of repetitions of each $b \in B$; number of repetitions written as $\mathcal{B}[b]$.*

E.g., for $B = \{\alpha, \beta, \gamma, \delta\}$, a "multiset" $\{\alpha, \beta, \beta, \beta, \delta, \delta\}$ would be written as $\mathcal{B} = (1, 3, 0, 2)$ where we assume an implicit ordering on B. We further denote:

- $b \in \mathcal{B}$: $\mathcal{B}[b] \neq 0$ ($\notin$ respectively),
- $\{b\}$ as a multiset $\mathcal{B}$: $\mathcal{B}[b] = 1$, $\mathcal{B}[\cdot] = 0$ otherwise,
- $(b_i; n_i)_{i=1}^{N}$ as a multiset $\mathcal{B}$: $\mathcal{B}[b_i] = n_i$, $\mathcal{B}[\cdot] = 0$ otherwise,
- $n \cdot \mathcal{B}$: scalar multiplication in $\mathbb{Z}^{|B|}$,
- $\mathcal{B}_1 \cup \mathcal{B}_2 := \mathcal{B}_1 + \mathcal{B}_2$, i.e., vector addition in $\mathbb{Z}^{|B|}$,
- $\sum_{b \in \mathcal{B}} (\cdot)_b := \sum_{b \in B} \mathcal{B}[b] \cdot (\cdot)_b$,
- $|\mathcal{B}| := \sum_{b \in \mathcal{B}} 1$, i.e., a sum of $\mathcal{B}[b]$, can be negative,
- $\mathcal{M}(B)\big|_{\mathbf{b}}$ where $\mathbf{b} \subseteq B$: a set of such $\mathcal{B} \in \mathcal{M}(B)$ where $\mathcal{B}[b] = 0$ for $b \in B \setminus \mathbf{b}$.
- for a finite set A, let $A^* = \bigcup_{n=0}^{\infty} A^n$,
- for a function $f \colon \mathbb{N} \to \mathbb{R}^+$, we say that f is
  - negligible if $\forall c > 0\ \exists k_c\ \forall k > k_c$ it holds $f(k) < k^{-c}$, also $f = negl(k)$,
  - overwhelming if $1 - f$ is negligible, $f \in \mathsf{OW}$,
- $a \leftarrow A$: uniformly random draw from the set A to the variable a,
- $a \div b$: integer division,
- $\hat{N} := \{1, 2, \ldots, N\}$,
- $\|\mathbf{x}\|_1$: ℓ_1-norm of the vector $\mathbf{x} = (x_1, \ldots, x_n)$, i.e., $\sum_{i=1}^{n} |x_i|$.

The VeraGreg Framework

A formal definition of the VeraGreg framework as per [13] is recalled in Appendix A, however, for a basic understanding, the following is sufficient. We also recall extended notions of security for VeraGreg which are based on those by Bellare et al. [2]; in particular we recommend to refresh NM [2, Definition 2.2].

Definition 2 (VeraGreg Scheme; informal). *By the* VeraGreg Scheme *we mean an instantiation of the* VeraGreg *framework, in particular, a* 5*-tuple of the following algorithms (simplified; see [13] for the full description):*

Init**.** *Generate keys for underlying ciphers (additively homomorphic encryption,* AHE*, and symmetric encryption,* SE*) and pick (large) random integers* $m_{1,2}$*. Output public key of* AHE *as* pk *and secret keys together with* $m_{1,2}$ *as* sk*.*

Grant**.** *Input data* d *and check its validity. Grant an ID* independent *on* d *and output it as* b*.*

$\mathsf{E_{sk}}$**.** *Encrypt input data* d *and granted ID* b *as*

$$c_b = \mathsf{E_{sk}}(b, d) = \mathrm{AHE}_{\mathsf{pk}}\big((\mathrm{SE}_{\mathsf{sk}}(b) \cdot m_1 + d) \cdot m_2\big). \tag{1}$$

$\mathsf{Add_{pk}}$**.** *Employ the homomorphic property and aggregate provided ciphertexts based on the list of ID's denoted by* $\mathcal{B}$ *(a multiset),*

$$\mathsf{Add_{pk}}\big(\mathcal{B}, (c_b)_{b\in\mathcal{B}}\big) = \bigoplus_{b\in\mathcal{B}} c_b \tag{2}$$

where $\oplus$ *denotes the ciphertext addition operation of* AHE*. Prevent possible inner overflow of* $m_{1,2}$ *or* AHE *plaintext (cf.* (1)*) by returning* $\bot$*.*

$\mathsf{D_{sk}}$**.** *Proceed as follows:*

```
1: function $\mathsf{D_{sk}}(\mathcal{B}, c)$
2:     if $\mathcal{B}$ is not policy-compliant then return $\bot$
3:     $\tilde{p} = \mathrm{AHE}^{-1}_{\mathsf{sk}}(c)$
4:     if $\tilde{p} \bmod m_2 \neq 0$ then return $\bot$
5:     $\tilde{b}_\Sigma = \tilde{p} \div (m_1 m_2)$
6:     $b_\Sigma = \sum_{b\in\mathcal{B}} \mathrm{SE}_{\mathsf{sk}}(b)$
7:     if $\tilde{b}_\Sigma \neq b_\Sigma$ then return $\bot$
8:     $\tilde{d} = (\tilde{p} \div m_2) \bmod m_1$
9:     return $\tilde{d}$
```

Here, policy-compliance (line 2) is evaluated with respect to a VeraGreg policy *(see [13] for details) which is a subset of* $\mathcal{M}(B)$ *– a subset of allowed multisets* $\mathcal{B}$*.*

Remark 1. The VeraGreg scheme by Definition 2 is somewhat homomorphic in the sense outlined in [9]; here we insist on the possibility of addition of at least 2^ν values before an overflow occurs, see [13] for details.

Definition 3 (LNM; informal). *In addition to the definition of Non-Malleability by Bellare et al. [2],* List Non-Malleability *(LNM) further requires that the output ciphertexts do not include the ID of the challenge within their lists.*

Definition 4 (LCCA2; informal). *In addition to the definition of Adaptive Chosen Ciphertext Attack by Bellare et al. [2],* List Adaptive Chosen Ciphertext Attack *(LCCA2) restricts the decryption oracle during the second phase: it refuses to decrypt any ciphertext that includes the ID of the challenge within its list.*

Theorem (LNM-LCCA2 $\iff$ IND-LCCA2, [13])**.** *A* VeraGreg *framework is LNM-LCCA2-secure if and only if it is IND-LCCA2-secure.*

Theorem (IND-CCA1$_{\mathrm{AHE}}$ $\Rightarrow$ IND-CCA1$_{\mathcal{V}}$, [13])**.** *Let $\mathcal{V}$ be a* VeraGreg *scheme. If its* AHE *is IND-CCA1-secure, $\mathcal{V}$ is also IND-CCA1-secure.*

3 Weak and Strong List Non-Malleability

In 2013, Armknecht et al. [1] have shown Paillier AHE to be IND-CCA1-secure. On the one hand, we obtain an IND-CCA1-secure VeraGreg scheme, on the other hand, IND-CCA1 does not tell anything about the VeraGreg verification feature—simply because it is already achieved by the underlying AHE. Hence we insist on stronger notions of security that cannot be achieved by a plain AHE and that take the verification feature into account. In Definition 2, we employed some countermeasures—namely secret $m_{1,2}$ and encryption of ID's—, let us discuss their necessity for the (prospective) stronger notions.

Proposition 1. *No matter whether $m_{1,2}$ are private or public, if* SE *were an identity mapping, LNM would not be achievable by the* VeraGreg *scheme.*

Proof. For a granting algorithm that grants ID's starting from 1 and increments them by 1, the adversary wins any LNM experiment as follows: she submits at least one encryption query and after obtaining the challenge (b^*, c^*), she answers with $(\{1, b^* - 1\}, c^*)$ together with an identity relation. Note that such an answer is accepted as a successful attack: indeed, b^* is not present in the list, the pair passes verification and the identity relation clearly holds. $\square$

Remark 2. Such a breach works also in the case the adversary knows the (anyhow) modified ID's, e.g., by a hash function, and is able to combine two distinct multisets of them to yield the same sum, cf. line 6 and 7 in D_{sk} in Definition 2.

Proposition 2. *If $m_{1,2}$ were public, it would be possible to modify the data inside a* VeraGreg *scheme ciphertext effectively.*

Proof. The adversary can encrypt d as $c = \mathrm{AHE}(d \cdot m_2)$ and add it by $\oplus$ to a VeraGreg ciphertext without being detected by the decryption algorithm. $\square$

In the scenario of Proposition 2, there does not appear to be any evidence that ease of data modification contradicts LNM. Indeed, it still appears to be hard to extract the data or modify the list, however, even data modification shall be avoided. This leads us to another and yet stronger notion of security.

The weakness of the definition of LNM (Definition 3) is that it does not accept a ciphertext, where only data has been modified, as a valid attack, although, intuitively, it shall be accepted. Only trivially constructed ciphertexts (i.e., those combining existing ciphertexts using the addition algorithm) shall not be accepted while any other (including that one identified in the proof of Proposition 2) shall be accepted as a valid attack. In Table 1, we provide a summary of differences between the original and the desired version of list non-malleability, referred to as *Weak List Non-Malleability* (WLNM, formerly LNM) and *Strong List Non-Malleability* (SLNM), respectively. Formal definition of SLNM follows.

Table 1. Comparison of trivial and rejected ciphertexts for different types of non-malleability. Here $\bar{c}_1$ denotes a malformed ciphertext with modified data part.

	Classical NM	VeraGreg WLNM	VeraGreg SLNM
Trivial	c^*	$(c^* \oplus c_1, \{b^*, b_1\})$ etc.	
Rejected	c^*	$(c^* \oplus c_1, \{b^*, b_1\})$ etc. $(c^* \oplus \bar{c}_1, \{b^*, b_1\})$ etc.	$(c^* \oplus c_1, \{b^*, b_1\})$ etc.

Definition 5 (Trivial Breaches). *The set of* Trivial Breaches *with respect to a challenge ID-ciphertext pair (b^*, c^*), denoted by $\mathcal{TB}(b^*, c^*)$, is a set of all list-ciphertext pairs which are computable from yet obtained ID-ciphertext pairs $(b_i, c_i)_{i=1}^n \ni (b^*, c^*)$, $\mathbf{b} = \{b_1, \ldots, b_n\}$, using* Add *Algorithm or re-randomization (if applicable), and include b^* in the list, i.e.,*

$$\mathcal{TB}(b^*, c^*) = \Big\{ \Big(\mathcal{B}, \mathsf{Add}_{\mathsf{pk}}\big(\mathcal{B}, (c_i)_{i=1}^n\big)\Big) \;\Big|\; \mathcal{B} \in \mathcal{M}(B)\big|_{\mathbf{b}}, b^* \in \mathcal{B} \Big\}. \tag{3}$$

The following definition is given in the format of the definition of NM by Bellare et al. [2, Definition 2.2] in order to simplify its understanding to the reader.

Definition 6 (Strong List Non-Malleability). *Let $\mathcal{V} = (\mathsf{Init}, \mathsf{Grant}, \mathsf{E}, \mathsf{Add}, \mathsf{D})$ be a* VeraGreg *framework and $A = (A_1, A_2)$ an adversary. For $\mathsf{atk} \in \{\mathsf{CPA}, \mathsf{CCA1}, \mathsf{LCCA2}\}$ and $\lambda \in \mathbb{N}$ let*

$$\mathbf{Adv}_{\mathcal{V},A}^{\mathsf{SLNM\text{-}atk}}(\lambda) = \Pr[\mathbf{Exp}_{\mathcal{V},A}^{\mathsf{SLNM\text{-}atk\text{-}1}}(\lambda) = 1] - \Pr[\mathbf{Exp}_{\mathcal{V},A}^{\mathsf{SLNM\text{-}atk\text{-}0}}(\lambda) = 1] \tag{4}$$

where, for $q \in \{0, 1\}$,

```
experiment $\mathbf{Exp}_{\mathcal{V},A}^{\text{SLNM-ATK-}q}(\lambda)$
    $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Init}(\lambda)$
    $(M, s) \leftarrow A_1^{\mathcal{E},\mathcal{D}}(\mathsf{pk})$
    $d_0, d_1 \leftarrow M$; $b^* \leftarrow \mathsf{Grant}_{\lambda,\mathbf{b}}(d_1)$; $c^* \leftarrow \mathsf{E}_{\mathsf{sk}}(b^*, d_1)$
    $\big(R, (\mathcal{B}^{(i)}, c^{(i)})_{i=1}^N\big) \leftarrow A_2^{\mathcal{E},\mathcal{D}^*}\big(M, s, (b^*, c^*)\big)$
    $\mathbf{d} \leftarrow \mathsf{D}_{\mathsf{sk}}(\mathcal{B}^{(i)}, c^{(i)})_{i=1}^N$
```

```
7:      if $\forall i \in \hat{N} \colon (\mathcal{B}^{(i)}, c^{(i)}) \notin \mathcal{TB}(b^*, c^*) \land \bot \notin \mathbf{d} \land R(d_q, \mathbf{d})$ then
8:          return $1$
9:      else
10:         return $0$
```

where

$$\mathcal{E}(d) = \big(b \leftarrow \mathsf{Grant}_{\lambda,\mathbf{b}}(d), c \leftarrow \mathsf{E}_{\mathsf{sk}}(b, d)\big)$$

and,

$$\begin{array}{lll}
\textit{if } \mathsf{atk} = \mathit{CPA}, & \textit{then } \mathcal{D}(\cdot) = \varepsilon & \textit{and } \mathcal{D}^*(\cdot) = \varepsilon, \\
\textit{if } \mathsf{atk} = \mathit{CCA1}, & \textit{then } \mathcal{D}(\cdot) = \mathsf{D}_{\mathsf{sk}}(\cdot) & \textit{and } \mathcal{D}^*(\cdot) = \varepsilon, \\
\textit{if } \mathsf{atk} = \mathit{LCCA2}, & \textit{then } \mathcal{D}(\cdot) = \mathsf{D}_{\mathsf{sk}}(\cdot) & \textit{and} \\
 & \mathcal{D}^*(\mathcal{B}, c) = \mathsf{D}_{\mathsf{sk}}(\mathcal{B}, c) & \textit{if } b^* \notin \mathcal{B}, \textit{ and} \\
 & \mathcal{D}^*(\mathcal{B}, c) = \bot & \textit{if } b^* \in \mathcal{B}.
\end{array}$$

We say that $\mathcal{V}$ is $\mathsf{SLNM\text{-}atk}$-secure if, for every polynomial $p(\cdot)$, the following holds: if A runs in time $p(\lambda)$, outputs $M \subseteq D$ sampleable in time $p(\lambda)$, and outputs a relation R computable in time $p(\lambda)$ for every $\lambda \in \mathbb{N}$, then $\mathbf{Adv}^{\mathsf{SLNM\text{-}atk}}_{\mathcal{V},A}(\cdot)$ is negligible.

Note 1. The differences between the classical NM and SLNM are in the encryption oracle access that is provided to the SLNM adversary, in the ciphertext format and, in particular, in the condition on line 7 of Experiment $\mathbf{Exp}^{\mathsf{SLNM\text{-}atk\text{-}}q}_{\mathcal{V},A}$: the original $y \notin \mathbf{y}$ is replaced with $(\mathcal{B}^{(i)}, c^{(i)}) \notin \mathcal{TB}(b^*, c^*)$ – both dealing with a trivial breach. Further recall that respective WLNM condition states

$$\forall i \in \hat{N} \colon b^* \notin \mathcal{B}^{(i)} \land \bot \notin \mathbf{d} \land R(d_q, \mathbf{d}). \qquad (\mathsf{WLNM})$$

Remark 3. The definition of SLNM (also that of NM by Bellare et al.) requires certain effort to understand its intended meaning, in particular the meaning of the condition on line 7. The condition combines

- a non-triviality and validity check, with
- a relation R provided by the adversary.

The goal of the relation R is to distinguish a non-trivial attack that *targets the data in the challenge* (i.e., d_1 encrypted in c^*) from a trivial attack that *blindly targets an uncontrolled subset of data.* Hence the relation should hold for $q = 1$ since d_1 was encrypted as the challenge c^*, cf. line 4, but it should not hold for $q = 0$ since d_0 was a randomly drawn piece of data intended for this check. It follows that an attack with $\mathbf{Adv} = 1$ is only possible if the condition

$$\forall i \in \hat{N} \colon (\mathcal{B}^{(i)}, c^{(i)}) \notin \mathcal{TB}(b^*, c^*) \land \bot \notin \mathbf{d} \qquad (5)$$

is always true, and the relation

$$R(d_q, \mathbf{d}) \qquad (6)$$

only holds for the challenge, i.e., for $q = 1$. In case of an attack where (5) is not satisfied or (6) holds always or never, it results in $\mathbf{Adv} = 0$. Hence, $\mathbf{Adv}$ represents the ratio of successful and unsuccessful attacks as outlined above.

In order to support the definition of SLNM, we show in the following theorem that it is indeed stronger than the original notion of list non-malleability (WLNM).

Theorem 1 ($\mathsf{SLNM\text{-}atk} \Rightarrow \mathsf{WLNM\text{-}atk}$). *If a* VeraGreg *framework $\mathcal{V}$ resists SLNM in an attack scenario, then it resists WLNM in the same attack scenario.*

Proof. In the sense of Remark 3, we show that any successful WLNM attack is also a successful SLNM attack, hence $\mathbf{Adv}_{\mathcal{V},A}^{\mathsf{SLNM\text{-}atk}} \geq \mathbf{Adv}_{\mathcal{V},A}^{\mathsf{WLNM\text{-}atk}} - negl(\lambda)$ where A is a WLNM adversary and the negligible term is present due to undefined behavior of VeraGreg in corner cases, cf. Definition 2. Note that WLNM only differs from SLNM by the set of rejected vectors, cf. Table 1. The WLNM set of rejected vectors is a superset of that of SLNM, hence it follows that all of the attack vectors accepted in the WLNM experiment are also accepted in the SLNM experiment which concludes the proof. $\square$

Corollary 1. *$\mathsf{SLNM\text{-}LCCA2} \Rightarrow \mathsf{WLNM\text{-}LCCA2} \iff \mathsf{IND\text{-}LCCA2}$.*

In particular, we obtained the so far strongest notion of security for the VeraGreg framework – $\mathsf{SLNM\text{-}LCCA2}$. Note that SLNM cannot be strengthened any more by the means of rejected breaches treated as trivial.

4 VeraGreg Decomposition

In order to study theoretical guarantees of particular VeraGreg instantiations, we suggest how the abstract VeraGreg framework may internally work. We decompose the encryption algorithm into components for which we define novel notions. By this approach, we delegate the "global" guarantees towards the components whose novel notions will appear to be more tangible. We begin the decomposition of VeraGreg encryption by encapsulating an encoding of the ID-data pair into an additively homomorphic ciphertext, cf. (1) in Definition 2.

Definition 7 (VeraGreg Encoding Framework). *Let* $(\mathsf{Init}, \mathsf{Grant}, \mathsf{E}, \mathsf{Add}, \mathsf{D})$ *be a* VeraGreg *framework,* $\mathrm{AHE}\colon P_A \to C$ *an additively homomorphic encryption scheme where* $\oplus$ *denotes its ciphertext addition operation,* $\mathsf{Enc}\colon K_S \times B \times D \to P_A$ *an encoding algorithm and* $\mathsf{Dec}\colon K_S \times \mathcal{M}(B) \times P_A \to D \cup \{\perp\}$ *a decoding algorithm. Let these algorithms satisfy*

- Init *further inits* AHE *with a key pair* $(\mathsf{pk}_A, \mathsf{sk}_A)$ *while it stores* pk_A *into* pk *and* sk*, and* sk_A *into* sk*,*
- $\mathsf{E}_{\mathsf{sk}}(b, d) = \mathrm{AHE}_{\mathsf{pk}_A}\big(\mathsf{Enc}_{\mathsf{sk}}(b, d)\big)$,
- $\mathsf{Add}_{\mathsf{pk}}\big(\mathcal{B}, (c_b)_{b\in\mathcal{B}}\big) = \bigoplus_{b\in\mathcal{B}} c_b$,
- $\mathsf{D}_{\mathsf{sk}}(\mathcal{B}, c) = \mathsf{Dec}_{\mathsf{sk}}\big(\mathcal{B}, \mathrm{AHE}_{\mathsf{sk}_A}^{-1}(c)\big)$.

We call the 6-tuple $(\mathsf{Init}, \mathsf{Grant}, \mathrm{AHE}, \mathsf{Enc}, \mathsf{Add}, \mathsf{Dec})$ *the* VeraGreg Encoding Framework (VGE)*.*

Lemma 1. *Let* $(\mathsf{Init}, \mathsf{Grant}, \mathrm{AHE}, \mathsf{Enc}, \mathsf{Add}, \mathsf{Dec})$ *be a VGE. Then for any valid set of ID-data pairs* $(b, d_b)_{b\in\mathbf{b}}$*,* $\mathbf{b} \subseteq B$*, any policy-compliant* $\mathcal{B} \in \mathcal{M}(B)\big|_{\mathbf{b}}$ *and a key pair* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Init}_\lambda$*, it holds*

$$\mathsf{Dec}_{\mathsf{sk}}\Big(\mathcal{B}, \sum_{b\in\mathcal{B}} \mathsf{Enc}_{\mathsf{sk}}(b, d_b)\Big) = \sum_{b\in\mathcal{B}} d_b. \tag{7}$$

Proof. By Definition 14 (in Appendix A) and Definition 7 we have

$$\mathsf{Dec}_{\mathsf{sk}}\Big(\mathcal{B}, \sum_{b\in\mathcal{B}} \mathsf{Enc}_{\mathsf{sk}}(b, d_b)\Big) = \mathsf{D}_{\mathsf{sk}}\Big(\mathcal{B}, \mathrm{AHE}_{\mathsf{pk}_A}\big(\sum_{b\in\mathcal{B}} \mathsf{Enc}_{\mathsf{sk}}(b, d_b)\big)\Big) =$$

$$= \mathsf{D}_{\mathsf{sk}}\Big(\mathcal{B}, \bigoplus_{b\in\mathcal{B}} \mathrm{AHE}_{\mathsf{pk}_A}\big(\mathsf{Enc}_{\mathsf{sk}}(b, d_b)\big)\Big) = \mathsf{D}_{\mathsf{sk}}\Big(\mathcal{B}, \bigoplus_{b\in\mathcal{B}} \mathsf{E}_{\mathsf{sk}}(b, d_b)\Big) =$$

$$= \mathsf{D}_{\mathsf{sk}}\Big(\mathcal{B}, \mathsf{Add}_{\mathsf{pk}}\big(\mathcal{B}, \mathsf{E}_{\mathsf{sk}}(b, d_b)_{b\in\mathcal{B}}\big)\Big) = \sum_{b\in\mathcal{B}} d_b.$$

□

Definition 8. *Let* $(\mathsf{Init}, \mathsf{Grant}, \mathrm{AHE}, \mathsf{Enc}, \mathsf{Add}, \mathsf{Dec})$ *be a VGE. Then for any valid set of ID-data pairs* $(b, d_b)_{b\in\mathbf{b}}$*,* $\mathbf{b} \subseteq B$*, any policy-compliant* $\mathcal{B} \in \mathcal{M}(B)\big|_{\mathbf{b}}$*,* $d_\Sigma \coloneqq \sum_{b\in\mathcal{B}} d_b$ *and a key pair* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Init}_\lambda$*, we define* Augmented Encoding

$$\mathsf{Enc}'_{\mathsf{sk}}(\mathcal{B}, d_\Sigma) \coloneqq \sum_{b\in\mathcal{B}} \mathsf{Enc}_{\mathsf{sk}}(b, d_b). \tag{8}$$

In the following lemma, we show that Enc' is well defined.

Lemma 2. *Under the assumptions of Definition 8, it holds*

$$\mathsf{Enc}'_{\mathsf{sk}}(\{b\}, d_b) = \mathsf{Enc}_{\mathsf{sk}}(b, d_b), \tag{9}$$

and

$$\mathsf{Dec}_{\mathsf{sk}}\Big(\mathcal{B}, \mathsf{Enc}'_{\mathsf{sk}}\big(\mathcal{B}, \sum_{b\in\mathcal{B}} d_b\big)\Big) = \mathsf{Dec}_{\mathsf{sk}}\Big(\mathcal{B}, \sum_{b\in\mathcal{B}} \mathsf{Enc}'_{\mathsf{sk}}(\{b\}, d_b)\Big) = \sum_{b\in\mathcal{B}} d_b, \tag{10}$$

i.e., Enc' *is an augmentation of* Enc *and it is homomorphic in the sense of Equation* (10)*:* $\mathsf{Enc}'_{\mathsf{sk}}\big(\mathcal{B}, \sum_{b\in\mathcal{B}} d_b\big)$ *decodes the same as* $\sum_{b\in\mathcal{B}} \mathsf{Enc}'_{\mathsf{sk}}(\{b\}, d_b)$*.*

Proof. Equation (9) holds by Definition 8. Equation (10) holds by Definition 8 and Equations (9) and (7). □

In the following definition, we introduce a non-malleability notion for Enc which aims to serve as a prospective guarantee of SLNM of a VGE.

Definition 9 (Encoding Non-Malleability). *Let* Init*,* Enc*,* Dec *be respective algorithms of a VGE* $\mathcal{V}$*,* Grant *a granting algorithm and let* $A = (A_1, A_2)$ *be an adversary. For* $\mathsf{atk} \in \{\mathit{CEA0}, \mathit{CEA1}, \mathit{CEA2}\}$ *and* $\lambda \in \mathbb{N}$ *let*

$$\mathbf{Adv}^{\mathit{ENM\text{-}atk}}_{\mathcal{V},A}(\lambda) = \Pr[\mathbf{Exp}^{\mathit{ENM\text{-}atk\text{-}1}}_{\mathcal{V},A}(\lambda) = 1] - \Pr[\mathbf{Exp}^{\mathit{ENM\text{-}atk\text{-}0}}_{\mathcal{V},A}(\lambda) = 1] \tag{11}$$

where, for $q \in \{0, 1\}$*,*

```
experiment $\mathbf{Exp}^{\text{ENM-ATK-}q}_{\mathcal{V},A}(\lambda)$
    $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Init}(\lambda)$
    $(M, s) \leftarrow A_1^{\mathcal{E},\mathcal{D}}(\mathsf{pk})$
    $d_0, d_1 \leftarrow M$; $b^* \leftarrow \mathsf{Grant}_{\lambda,\mathbf{b}}(d_1)$; $e^* \leftarrow \mathsf{Enc}_{\mathsf{sk}}(b^*, d_1)$
    $\big(R, (\mathcal{B}^{(i)}, \mathcal{B}_{\Sigma}^{(i)}, e^{(i)})_{i=1}^{N}\big) \leftarrow A_2^{\mathcal{E},\mathcal{D}^*}(M, s, b^*)$
    $\mathbf{d} \leftarrow \mathsf{Dec}_{\mathsf{sk}}\big(\mathcal{B}^{(i)}, \sum_{b \in \mathcal{B}_{\Sigma}^{(i)}} e_b + e^{(i)}\big)_{i=1}^{N}$
    if $\forall i \in \hat{N}\colon (e^{(i)} \neq 0 \vee \mathcal{B}^{(i)} \neq \mathcal{B}_{\Sigma}^{(i)}) \wedge \perp \notin \mathbf{d} \wedge R(d_q, \mathbf{d})$ then
        return $1$
    else
        return $0$
```

where

$\mathcal{E}(d) = \mathsf{Grant}_{\lambda,\mathbf{b}}(d) \to b$

while it computes and keeps respective $e_b = \mathsf{Enc}_{\mathsf{sk}}(b, d)$

and,

$$\begin{array}{lll}
\textit{if } \mathsf{atk} = \mathsf{CEA0}, & \textit{then } \mathcal{D}(\cdot) = \varepsilon & \textit{and } \mathcal{D}^*(\cdot) = \varepsilon, \\
\textit{if } \mathsf{atk} = \mathsf{CEA1}, & \textit{then } \mathcal{D}(\mathcal{B}, \mathcal{B}_{\Sigma}, e) = \mathsf{Dec}_{\mathsf{sk}}(\mathcal{B}, \sum_{b \in \mathcal{B}_{\Sigma}} e_b + e) & \textit{and } \mathcal{D}^*(\cdot) = \varepsilon, \\
\textit{if } \mathsf{atk} = \mathsf{CEA2}, & \textit{then } \mathcal{D}(\mathcal{B}, \mathcal{B}_{\Sigma}, e) = \mathsf{Dec}_{\mathsf{sk}}(\mathcal{B}, \sum_{b \in \mathcal{B}_{\Sigma}} e_b + e) & \textit{and} \\
 & \mathcal{D}^*(\mathcal{B}, \mathcal{B}_{\Sigma}, e) = \mathsf{Dec}_{\mathsf{sk}}(\mathcal{B}, \sum_{b \in \mathcal{B}_{\Sigma}} e_b + e) & \textit{if } b^* \notin \mathcal{B}, \textit{ and} \\
 & \mathcal{D}^*(\mathcal{B}, \mathcal{B}_{\Sigma}, e) = \perp & \textit{if } b^* \in \mathcal{B}.
\end{array}$$

We say that $\mathcal{V}$'s encoding is ENM-atk-secure if, for every polynomial $p(\cdot)$, the following holds: if A runs in time $p(\lambda)$, outputs $M \subseteq D$ sampleable in time $p(\lambda)$, and outputs a relation R computable in time $p(\lambda)$ for every $\lambda \in \mathbb{N}$, then $\mathbf{Adv}^{\textsf{ENM-atk}}_{\mathcal{V},A}(\cdot)$ is negligible

Note 2. In the previous definition, CEA stands for *Chosen Encoding Attack*.

Remark 4. In the ENM experiment, the adversary gets *absolutely no information* related to actual encodings or data – she only gets ID's which are required to be independent on actual data, cf. Definition 2. Hence, from the point of view of the adversary, it appears like she were given black boxes labelled by ID's.

Theorem 2 (SLNM-atk ⇒ ENM-atk′). *If a VGE is SLNM-atk-secure, then its encoding is ENM-atk'-secure, for CPA–CEA0, CCA1–CEA1 and LCCA2–CEA2 pairs of atk–atk'.*

Proof. Find the proof in Appendix B.1. □

Before we state a restricted variant of the opposite implication, we define an AHE scheme which can be perceived to operate with black boxes, cf. Remark 4.

Definition 10 (Perfect AHE). Perfect Additively Homomorphic Encryption scheme (AHE*) *is an* AHE *scheme where let λ, $\delta \in \mathbb{N}$ be the security and data space parameter, respectively, $2^\delta \ll 2^\lambda$, $D = \{0,1\}^\delta$ the plaintext space, $C = \mathcal{M}(\{0,1\}^\lambda)$ the ciphertext space (multisets of λ-bit strings) and DB a database/mapping, $DB\colon \{0,1\}^\lambda \to D \cup \{\perp\}$ initialized as $DB(\cdot) = \perp$. We describe the encryption and decryption oracles $\mathcal{E}$, $\mathcal{D}$, respectively, and the ciphertext addition operation $\bigoplus$.*

Encryption Oracle.

```
function $\mathcal{E}_{DB}(d)$
    $c \leftarrow \{0,1\}^\lambda$
    $DB(c) \leftarrow d$
    return $\mathcal{C} \leftarrow \{c\}$
```

Addition $\bigoplus$.

```
function $\bigoplus((\mathcal{C}_i)_{i=1}^N, (n_i)_{i=1}^N)$
    return $\mathcal{C} \leftarrow \bigcup_{i=1}^N n_i \cdot \mathcal{C}_i$
```

Decryption Oracle.

```
function $\mathcal{D}_{DB}(\mathcal{C})$
    if $\exists c \in \mathcal{C}, c \notin DB$ then return $\perp$
    decompose the multiset $\mathcal{C}$ into $(c_i, n_i)_{i=1}^N$
    $d \leftarrow 0$
    for $i = 1 \ldots N$ do
        $d_i \leftarrow DB(c_i)$
        if $d_i = \perp$ then return $\perp$
        $d \leftarrow d + n_i \cdot d_i$
    return $d$
```

Theorem 3 (ENM-atk′ ⇒ SLNM-atk$_{\mathrm{AHE}^*}$). *Let a VGE $\mathcal{V}$ use the perfect* AHE*. If its encoding is ENM-atk'-secure, then it is SLNM-atk-secure for CEA0–CPA, CEA1–CCA1 and CEA2–LCCA2 pairs of atk'–atk.*

Proof. Find the proof in Appendix B.2. □

Corollary 2. *If a VGE employs* AHE**, it is SLNM-atk-secure if and only if its encoding is ENM-atk'-secure, for respective pairs of atk–atk'.*

In order to get the definitions of ENM and SLNM yet closer to each other, we define a variant of the ENM experiment where the adversary has an access to actual ciphertexts, in addition to ID's, i.e., similar to the SLNM experiment.

Definition 11 (ENM-atk$_{\mathrm{AHE}}$). *Let* AHE *be an additively homomorphic encryption scheme and* $(\mathsf{pk}_A, \mathsf{sk}_A)$ *its keypair. If in an ENM-atk experiment (as per Definition 9), atk* $\in \{\mathit{CEA0}, \mathit{CEA1}, \mathit{CEA2}\}$*, the adversary is further given* pk_A *(or* AHE *encryption and addition oracles), the encryption oracle* $\mathcal{E}$ *returns in addition* $c_b = \mathrm{AHE}(e_b)$*, and the decryption oracle* $\mathcal{D}$ *works instead as follows:*

$$\mathcal{D}(\mathcal{B}, c) = \mathsf{Dec}_{\mathsf{sk}}\big(\mathcal{B}, \mathrm{AHE}^{-1}_{\mathsf{sk}_A}(c)\big), \tag{12}$$

we denote this experiment as ENM-atk$_{\mathrm{AHE}}$.

Lemma 3 (ENM-atk$_{\mathrm{AHE}}$ ⇒ ENM-atk $\iff$ ENM-atk$_{\mathrm{AHE}^*}$). *Let* AHE *be an additively homomorphic encryption scheme,* AHE* *the perfect additively homomorphic encryption and atk* $\in \{\mathit{CEA0}, \mathit{CEA1}, \mathit{CEA2}\}$*. ENM-atk*$_{\mathrm{AHE}}$ *security implies ENM-atk security, which is equivalent to ENM-atk*$_{\mathrm{AHE}^*}$ *security.*

Proof. ⇒: In the ENM-atk$_{\mathrm{AHE}}$ experiment, the adversary has additional information, as opposed to the original ENM-atk experiment.

⇔: In the ENM-atk$_{\mathrm{AHE}^*}$ experiment, the adversary only has an access to AHE* ciphertexts in addition to the ENM-atk experiment. However, these are effectively random values, hence she cannot make any advantage of it. □

To conclude the first VeraGreg decomposition step, we conjecture that, providing the adversary with $\mathsf{IND\text{-}CCA1}$-secure AHE ciphertexts instead of AHE* "black boxes", the opposite implication in Lemma 3 and a variant of Theorem 3 hold. Note that such ciphertexts should also only allow for encryption of custom plain data and ciphertext addition, cf. Definition 9 and 10, hence we consider these conjectures reasonable. As a result, this leads to a conjecture on an equivalence of SLNM and ENM of a VGE that employs an $\mathsf{IND\text{-}CCA1}$-secure AHE.

Conjecture 1 ($\mathsf{ENM\text{-}atk}_{\mathrm{AHE}^*} \Rightarrow \mathsf{ENM\text{-}atk}_{\mathsf{IND\text{-}CCA1}}$). Let AHE be $\mathsf{IND\text{-}CCA1}$-secure. If a VGE encoding is $\mathsf{ENM\text{-}atk}_{\mathrm{AHE}^*}$-secure, then it is $\mathsf{ENM\text{-}atk}_{\mathrm{AHE}}$-secure, for $\mathsf{atk} \in \{\mathsf{CEA0}, \mathsf{CEA1}, \mathsf{CEA2}\}$.

Conjecture 2 ($\mathsf{ENM\text{-}atk}'_{\mathsf{IND\text{-}CCA1}} \Rightarrow \mathsf{SLNM\text{-}atk}$). Let AHE be $\mathsf{IND\text{-}CCA1}$-secure. If a VGE using AHE has an $\mathsf{ENM\text{-}atk}'_{\mathrm{AHE}}$-secure encoding, then it is $\mathsf{SLNM\text{-}atk}$-secure, for respective pairs of atk'–atk.

Conjecture 3 ($\mathsf{SLNM\text{-}atk} \iff \mathsf{ENM\text{-}atk}'$). Let AHE be $\mathsf{IND\text{-}CCA1}$-secure. A VGE using AHE is $\mathsf{SLNM\text{-}atk}$-secure if and only if it has an $\mathsf{ENM\text{-}atk}$'-secure encoding, for respective pairs of atk–atk'.

5 Verification using Sum Comparison

In the previous section, we decomposed VeraGreg encryption into encoding encapsulated by AHE. In this section, we focus on encoding, in particular, we specify the verification procedure. First, we suggest to aggregate the ID-related information by addition. Remind Proposition 1 which (in certain sense) calls for an unpredictable modification of ID's. In the following, we will model the unpredictable modification by a *Random Oracle* (RO). Recall that a random oracle $\mathrm{RO}\colon X \to Y$ is basically a randomly drawn function $f\colon X \to Y$, originally formulated by Bellare et al. [3].

Definition 12 (VeraGreg Internal Encoding Framework). *Let* $(\mathsf{Init}, \mathsf{Grant}, \mathrm{AHE}, \mathsf{Enc}, \mathsf{Add}, \mathsf{Dec})$ *be a* VeraGreg *encoding framework,* $\mathrm{RO}\colon B \to (R, +)$ *a random oracle,* $\mathsf{Inc}\colon K_S \times R \times D \to P_A$ *an internal encoding algorithm and* $\mathsf{Idc}\colon K_S \times P_A \to R \times D \cup \{\bot\}$ *an internal decoding algorithm. Let these algorithms satisfy*

- $\mathsf{Enc}_{\mathsf{sk}}(b, d) = \mathsf{Inc}_{\mathsf{sk}}\big(\mathrm{RO}(b), d\big)$*, and*
- $\mathsf{Dec}_{\mathsf{sk}}\big(\mathcal{B}, e\big)$ *proceeds as follows:*

```
function $\mathsf{Dec}_{\mathsf{sk}}(\mathcal{B}, e)$
    if $\mathcal{B}$ is not policy-compliant then return $\bot$
    if $\mathsf{Idc}_{\mathsf{sk}}(e) = \bot$ then return $\bot$
    $(r, d) \leftarrow \mathsf{Idc}_{\mathsf{sk}}(e)$
    if $r \neq \sum_{b \in \mathcal{B}} \mathrm{RO}(b)$ then return $\bot$
    return $d$
```

We call the 7*-tuple* $(\mathsf{Init}, \mathsf{Grant}, \mathrm{AHE}, \mathrm{RO}, \mathsf{Inc}, \mathsf{Add}, \mathsf{Idc})$ *the* VeraGreg Internal Encoding Framework (VGIE)*.*

Lemma 4. *Let* $(\mathsf{Init}, \mathsf{Grant}, \mathrm{AHE}, \mathrm{RO}, \mathsf{Inc}, \mathsf{Add}, \mathsf{Idc})$ *be a VGIE. Then for any valid set of ID-data pairs* $(b, d_b)_{b \in \mathbf{b}}$*,* $\mathbf{b} \subseteq B$*,* $r_b \coloneqq \mathrm{RO}(b)$*, any* $\mathcal{B} \in \mathcal{M}(B)\big|_{\mathbf{b}}$ *and a key pair* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Init}_\lambda$*, it holds*

$$\mathsf{Idc}_{\mathsf{sk}}\Big(\sum_{b\in\mathcal{B}} \mathsf{Inc}_{\mathsf{sk}}(r_b, d_b)\Big) = \Big(\sum_{b\in\mathcal{B}} r_b, \sum_{b\in\mathcal{B}} d_b\Big). \tag{13}$$

Proof. By Definition 12 and Lemma 1 we have

$$\mathsf{Idc}_{\mathsf{sk}}\Big(\sum_{b\in\mathcal{B}} \mathsf{Inc}_{\mathsf{sk}}(r_b, d_b)\Big) = \mathsf{Idc}_{\mathsf{sk}}\Big(\sum_{b\in\mathcal{B}} \mathsf{Enc}_{\mathsf{sk}}(b, d_b)\Big) =$$
$$= \Big(\sum_{b\in\mathcal{B}} \mathrm{RO}(b), \mathsf{Dec}_{\mathsf{sk}}\big(\mathcal{B}, \sum_{b\in\mathcal{B}} \mathsf{Enc}_{\mathsf{sk}}(b, d_b)\big)\Big) = \Big(\sum_{b\in\mathcal{B}} r_b, \sum_{b\in\mathcal{B}} d_b\Big).$$

□

Corollary 3. Inc *is homomorphic in the sense of*

$$\mathsf{Idc}_{\mathsf{sk}}\Big(\sum_{b\in\mathcal{B}} \mathsf{Inc}_{\mathsf{sk}}(r_b, d_b)\Big) = \Big(\sum_{b\in\mathcal{B}} r_b, \sum_{b\in\mathcal{B}} d_b\Big) = \mathsf{Idc}_{\mathsf{sk}}\Big(\mathsf{Inc}_{\mathsf{sk}}\big(\sum_{b\in\mathcal{B}} r_b, \sum_{b\in\mathcal{B}} d_b\big)\Big), \tag{14}$$

i.e., a sum of Inc*'s decodes the same as* Inc *of respective sums.*

Proof. Let b_Σ be a valid ID, let us enforce $\mathrm{RO}(b_\Sigma) = \sum_{b\in\mathcal{B}} r_b \eqqcolon r_\Sigma$ and let $d_\Sigma = \sum_{b\in\mathcal{B}} d_b$. The claim follows by Lemma 4 with (b_Σ, d_Σ) and $\mathcal{B} = \{b_\Sigma\}$. □

Note that in our definition of the VeraGreg internal encoding framework, the $\mathsf{Inc}/\mathsf{Idc}$ algorithms still implement a portion of VeraGreg security. Indeed, it should be still impossible to compute a valid encoding of any piece of data. Let us formulate this property in the following definition (cf. Definition 9).

Definition 13 (Internal Encoding Non-Malleability). *Let* Init*,* Inc*,* Idc *be respective algorithms of a VGIE* $\mathcal{V}$*,* RO *its random oracle,* Grant *a granting algorithm and let* $A = (A_1, A_2)$ *be an adversary. For* $\mathsf{atk} \in \{\mathit{CIA0}, \mathit{CIA1}, \mathit{CIA2}\}$ *and* $\lambda \in \mathbb{N}$ *let*

$$\mathbf{Adv}_{\mathcal{V},A}^{\textsf{\textit{INM-atk}}}(\lambda) = \Pr[\mathbf{Exp}_{\mathcal{V},A}^{\textsf{\textit{INM-atk}}\text{-}1}(\lambda) = 1] - \Pr[\mathbf{Exp}_{\mathcal{V},A}^{\textsf{\textit{INM-atk}}\text{-}0}(\lambda) = 1] \tag{15}$$

where, for $q \in \{0, 1\}$*,*

```
experiment $\mathbf{Exp}_{\mathcal{V},A}^{\text{INM-ATK-}q}(\lambda)$
    $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Init}(\lambda)$
    $(M, s) \leftarrow A_1^{\mathcal{E},\mathcal{D}}(\mathsf{pk})$
    $d_0, d_1 \leftarrow M$; $b^* \leftarrow \mathsf{Grant}_{\lambda,\mathbf{b}}(d_1)$; $r^* \leftarrow \mathrm{RO}(b^*)$; $e^* \leftarrow \mathsf{Inc}_{\mathsf{sk}}(r^*, d_1)$
    $\big(R, (\mathcal{B}_\Sigma^{(i)}, e^{(i)})_{i=1}^N\big) \leftarrow A_2^{\mathcal{E},\mathcal{D}^*}(M, s, b^*)$
    $(\mathbf{r}, \mathbf{d}) \leftarrow \mathsf{Idc}_{\mathsf{sk}}\big(\sum_{b\in\mathcal{B}_\Sigma^{(i)}} e_b + e^{(i)}\big)_{i=1}^N$
    if $\forall i \in \hat{N}\colon e^{(i)} \neq 0 \wedge \bot \notin (\mathbf{r}, \mathbf{d}) \wedge R(d_q, \mathbf{d})$ then
        return 1
    else
        return $0$
```

where

$\mathcal{E}(d) = b \leftarrow \mathsf{Grant}_{\lambda,\mathbf{b}}(d)$

while it computes and keeps $r = \mathrm{RO}(b)$ *and respective* $e_b = \mathsf{Inc}_{\mathsf{sk}}(r, d)$

and,

if $\mathsf{atk} = \mathsf{CIA0}$,	*then* $\mathcal{D}(\cdot) = \varepsilon$	*and* $\mathcal{D}^*(\cdot) = \varepsilon$,
if $\mathsf{atk} = \mathsf{CIA1}$,	*then* $\mathcal{D}(\mathcal{B}_\Sigma, e) = \mathsf{Idc}_{\mathsf{sk}}(\sum_{b \in \mathcal{B}_\Sigma} e_b + e)$	*and* $\mathcal{D}^*(\cdot) = \varepsilon$,
if $\mathsf{atk} = \mathsf{CIA2}$,	*then* $\mathcal{D}(\mathcal{B}_\Sigma, e) = \mathsf{Idc}_{\mathsf{sk}}(\sum_{b \in \mathcal{B}_\Sigma} e_b + e)$	*and*
	$\mathcal{D}^*(\mathcal{B}_\Sigma, e) = \mathsf{Idc}_{\mathsf{sk}}(\sum_{b \in \mathcal{B}_\Sigma} e_b + e)$	*if* $b^* \notin \mathcal{B}_\Sigma$, *and*
	$\mathcal{D}^*(\mathcal{B}_\Sigma, e) = \perp$	*otherwise.*

We say that $\mathcal{V}$*'s internal encoding is* INM-atk*-secure if, for every polynomial* $p(\cdot)$*, the following holds: if* A *runs in time* $p(\lambda)$*, outputs* $M \subseteq D$ *sampleable in time* $p(\lambda)$*, and outputs a relation* R *computable in time* $p(\lambda)$ *for every* $\lambda \in \mathbb{N}$*, then* $\mathbf{Adv}^{\mathsf{INM\text{-}atk}}_{\mathcal{V},A}(\cdot)$ *is negligible*

Note 3. In the definition above, CIA stands for *Chosen Internal Encoding Attack.*

Before we put INM into context with ENM, we study a combination of the somewhat homomorphic property (cf. Remark 1) with possible INM adversary's access to the RO. First, let us formulate a problem to be hard in order to prevent the situation outlined in Remark 2, i.e., distinct multisets of known modified ID's result in equal control sums.

Problem 1 (Sum of Randoms, SoR*).* Let λ be a security parameter and ν such that $poly(\lambda) \ll 2^\nu \ll 2^\lambda$. Given a random oracle $\mathrm{RO}_\lambda \colon B \to (R, +)$, $|R| \geq 2^\lambda$, with answers stored in a vector $\mathbf{r}$, *either* find an integer vector $\mathbf{w} \neq \mathbf{0}$ (referred to as the *vector solution*) such that $\mathbf{w} \cdot \mathbf{r} = 0$ and $\|\mathbf{w}\|_1 < 2^\nu$, *or* answer that such a vector does not exist. We refer to this problem as the *Sum of Randoms*, denoted as $\mathsf{SoR}_{\mathrm{RO}_\lambda}$. We say it is *intractable* if it only has a vector solution with negligible probability or it is computationally infeasible to find a vector solution.

Note 4. In order SoR to be tractable, $\mathbf{w}$ must be of a polynomial dimension. The condition $\|\mathbf{w}\|_1 < 2^\nu$ is present due to the somewhat restriction, cf. Remark 1.

Proposition 3. *Let there exist a polynomial* p *such that the* $\mathsf{SoR}_{\mathrm{RO}_\lambda}$ *problem has a non-empty set of vector solutions of dimension at most* $p(\lambda)$*, with non-negligible probability. Let further* $\mathcal{S}$ *be an oracle with an access to* RO_λ *that either, with non-negligible probability and after at most* $p(\lambda)$ *queries to* RO_λ*, returns a vector solution, or answers* $\perp$*. Given an access to* $\mathcal{S}$ *and* RO_λ*, no VGIE (even a somewhat homomorphic) can satisfy any kind of* WLNM- *or* LCCA2*-security.*

Proof. Let ($\mathsf{Init}, \mathsf{Grant}, \mathrm{AHE}, \mathrm{RO}_\lambda, \mathsf{Inc}, \mathsf{Add}, \mathsf{Idc}$) be a VGIE and $\mathbf{Exp}$ a WLNM- or LCCA2-type experiment. Since we do not control the granting algorithm, we cannot provide $\mathcal{S}$ with a direct access to RO_λ. Instead, with each query $b_\mathcal{S}$ of $\mathcal{S}$, we either look it up in our database (in case it has already been asked), or query the $\mathcal{E}$ oracle in $\mathbf{Exp}$ on, e.g., $d = 0$, to obtain fresh b, reply with $r \leftarrow \mathrm{RO}_\lambda(b)$

and store $(b_{\mathcal{S}}, r)$ in the database. Note that due to the uniformly random nature of the random oracle, this setup is equivalent to the case where $\mathcal{S}$ has a direct access to RO_λ, up to a polynomial slowdown.

With non-negligible probability and after at most $p(\lambda)$ queries, $\mathcal{S}$ returns a vector solution $\mathbf{w}$, clearly $\dim \mathbf{w} \leq p(\lambda)$. Note that for such $\mathbf{w}$, it occurs $\mathbf{w}[b^*] \neq 0$ with at least $\frac{1}{p(\lambda)}$ probability which keeps the overall probability non-negligible. It follows that with non-negligible probability, we can exploit $\mathbf{w}$ to replace the challenge ID b^* in the list $\mathcal{B}$ and answer trivially what we are supposed to; cf. the idea of the proof of Proposition 1. □

Remark 5. Due to the claim of Proposition 3, we insist on the assumption that SoR is intractable. Below we summarize a couple of related ideas and thoughts:

- if we omit $\|\mathbf{w}\|_1 < 2^\nu$, we have the following solutions

$$\mathbf{w}^{(\mathbf{k})} = \left(0, \ldots, 0, \frac{r_{k+1}}{GCD(r_k, r_{k+1})}, -\frac{r_k}{GCD(r_k, r_{k+1})}, 0, \ldots, 0\right)$$

  which form a basis of a lattice $\mathbf{W}$ where it holds $\mathbf{w} \cdot \mathbf{r} = \mathbf{0}$, $\forall \mathbf{w} \in \mathbf{W}$,
- $\mathbf{W}$ is a sublattice of the lattice of all integer solutions to $\mathbf{w} \cdot \mathbf{r} = \mathbf{0}$, however, there is no guarantee that they are equal,
- in modular lattices (typically in $\mathbb{Z}_p$, p prime), the problem of finding the shortest vector is believed to be hard on average (aka. the Shortest Vector Problem, SVP).

In the following theorem and its corollary, we finally put INM and ENM into context. We also cover the case of a somewhat homomorphic VGIE with an INM adversary with an access to the RO.

Theorem 4 (INM-atk $\Rightarrow$ ENM-atk′). *Let $\mathcal{V} = (\mathsf{Init}, \mathsf{Grant}, \mathrm{AHE}, \mathrm{RO}, \mathsf{Inc}, \mathsf{Add}, \mathsf{Idc})$ be a VGIE. If the internal encoding of $\mathcal{V}$ is INM-atk-secure, then its encoding is ENM-atk'-secure, for CIA0–CEA0, CIA1–CEA1 and CIA2–CEA2 pairs of atk–atk'.*

Proof. Find the proof in Appendix B.3. □

Corollary 4. *Assuming that* SoR *is intractable, Theorem 4 holds also for a somewhat homomorphic VGIE where the INM-atk adversary has, in addition, an access to the* RO*.*

Proof. Find the proof in Appendix B.4. □

Finally, we conjecture that the opposite implication in Theorem 4 holds, too. As a result of previous theorems and conjectures, this leads to a conjecture on an equivalence of SLNM and INM of a VGIE that employs an IND-CCA1-secure AHE, or, by Corollary 4, under the assumption of SoR intractability and the somewhat homomorphic property, even if RO is available to the adversary. This equivalence poses the ultimate goal of our decomposition effort – the VERA-GREG security would rely solely on the security of underlying constructs. We summarize our theorems, lemmas and conjectures in Figure 1.

Conjecture 4 (ENM-atk′ ⇒ INM-atk). If Enc of a VGIE is ENM-atk'-secure, then its Inc is INM-atk-secure for respective pairs of atk–atk'.

Conjecture 5 (SLNM-atk ⟺ INM-atk′). Let AHE be IND-CCA1-secure. A VGIE using AHE is SLNM-atk-secure if and only if its Inc is INM-atk'-secure for respective pairs of atk–atk'.

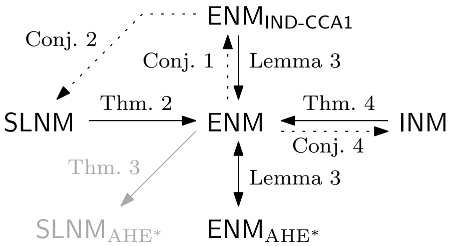

Fig. 1. A simplified summary of our results. Proven implications are drawn with solid lines, conjectures with dotted lines.

Remark 6. A cryptographic hash function (CHF) or a pseudorandom function (PRF) can be modelled as a random oracle and vice versa, for a comprehensive reading we refer to Canetti et al. [6]. In the definition of the VeraGreg scheme (Definition 2), we employed a symmetric encryption SE for an unpredictable modification of ID's, cf. (1). Here, SE is instantiated with a secret key, i.e., it is a private PRF which can be modelled as a random oracle that is not available to the adversary. However, with the assumptions of Corollary 4 (the VeraGreg scheme is indeed somewhat homomorphic), we can provide the random oracle to the adversary, i.e., it can be instantiated as a CHF instead. Hence we suggest

$$\mathsf{E}_{\mathsf{sk}}(b, d) = \mathrm{AHE}\big((h(b) \cdot m_1 + d) \cdot m_2\big) \tag{16}$$

where $h(\cdot)$ is a CHF. In other cases, PRF must be still employed.

Conclusion

As the major output of this paper, we consider the novel notions of security, in particular the "global" strong list non-malleability—for the abstract VeraGreg framework—and the "local" internal encoding non-malleability—for the structured VeraGreg internal encoding framework. Based on our theorems, we conjectured their equivalence under reasonable assumptions on the components of the structured framework. Finally, we identified a lattice problem to be hard in order to simplify the original somewhat homomorphic VeraGreg scheme.

Future Directions

We aim to prove the conjectures stated in this paper – 5 conjectures in total, two of them only as prospective corollaries. Although in some cases they are similar to finished proofs, these resisted our effort.

As another research topic, the SoR problem attracts our attention, in particular its relation to the famous Shortest Vector Problem (SVP) in lattice theory.

Finally, we focus on another instantiations of the Inc mapping – at the moment we work with Inc based on the encryption algorithm introduced in [13], i.e., $\mathsf{Inc}_{m_{1,2}}(r,d) = \big(r \cdot m_1 + d\big) \cdot m_2$ where $m_{1,2}$ are secret constants. For any future Inc mapping, we will particularly focus on its prospective INM security as introduced in this paper.

Acknowledgment

We would like to thank to Pascal Paillier for a brief yet very useful feedback.

Appendix

A Definition of the VeraGreg Framework

Definition 14 (VeraGreg Framework). *Let D denote an additive Abelian group—the data space—, $\lambda \in \mathbb{N}$ the security parameter, B the set of ID's, C the ciphertext space, K_P, K_S the public and secret key space, respectively.* VeraGreg Framework *is a* 5*-tuple of PPT algorithms* $(\mathsf{Init}, \mathsf{Grant}, \mathsf{E}, \mathsf{Add}, \mathsf{D})$*,*

- $\mathsf{Init}\colon \{1\}^* \to K_P \times K_S$,
- $\mathsf{Grant}\colon \{1\}^* \times B^* \times D \to B^* \times B \cup \{\bot\}$,
- $\mathsf{E}\colon K_S \times B \times D \to C$,
- $\mathsf{Add}\colon K_P \times \mathbb{Z}^{|B|} \times C^* \to C$,
- $\mathsf{D}\colon K_S \times \mathbb{Z}^{|B|} \times C \to D \cup \{\bot\}$,

for which it holds: $\forall n \in \mathbb{N}$, $\forall (d_i)_{i=1}^n \in D^n$ and respective valid $(b_i)_{i=1}^n =: \mathbf{b}$ granted by Grant_λ, $\forall \mathcal{B} \in \mathcal{M}(B)\big|_{\mathbf{b}}$ and a key pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Init}_\lambda$,

1. *if $\mathcal{B}$ is policy-compliant,*

$$\Pr\Big[\mathsf{D}_{\mathsf{sk}}\Big(\mathcal{B}, \mathsf{Add}_{\mathsf{pk}}\big(\mathcal{B}, \mathsf{E}_{\mathsf{sk}}(b_i, d_i)_{i=1}^n\big)\Big) = \sum_{i=1}^{n} \mathcal{B}[b_i] \cdot d_i\Big] \in \mathsf{OW}_\lambda, \tag{17}$$

   i.e., the encryption is additively homomorphic,

2. *if $\mathcal{B}$ is not policy-compliant,*

$$\Pr\Big[\mathsf{D}_{\mathsf{sk}}\Big(\mathcal{B}, \mathsf{Add}_{\mathsf{pk}}\big(\mathcal{B}, \mathsf{E}_{\mathsf{sk}}(b_i, d_i)_{i=1}^n\big)\Big) = \bot\Big] \in \mathsf{OW}_\lambda, \tag{18}$$

   i.e., policy-incompliant list is discarded, *3.* $\forall \mathcal{B}' \in \mathbb{Z}^{|B|}$, $\mathcal{B}' \neq \mathcal{B}$,

$$\Pr\Big[\mathsf{D}_{\mathsf{sk}}\Big(\mathcal{B}', \mathsf{Add}_{\mathsf{pk}}\big(\mathcal{B}, \mathsf{E}_{\mathsf{sk}}(b_i, d_i)_{i=1}^{n}\big)\Big) = \bot\Big] \in \mathsf{OW}_\lambda, \tag{19}$$

i.e., the framework detects any list forgery,

4. otherwise,

$$\Pr\big[\mathsf{D}_{\mathsf{sk}}(\cdot,\cdot) = \bot\big] \in \mathsf{OW}_\lambda, \tag{20}$$

i.e., any invalid ciphertext is detected.

B Security Reductions

In the following proofs, we omit the negligible term that is present due to undefined behavior of VeraGreg in corner cases, cf. the proof of Theorem 1.

B.1 Proof of Theorem 2

Theorem 2 ($\mathsf{SLNM\text{-}atk} \Rightarrow \mathsf{ENM\text{-}atk}'$)**.** *If a VGE is* $\mathsf{SLNM\text{-}atk}$*-secure, then its encoding is* $\mathsf{ENM\text{-}atk}$*'-secure, for* CPA*–*$\mathsf{CEA0}$*,* $\mathsf{CCA1}$*–*$\mathsf{CEA1}$ *and* $\mathsf{LCCA2}$*–*$\mathsf{CEA2}$ *pairs of* atk*–*atk*'.*

Proof. We show that $\mathbf{Adv}^{\mathsf{SLNM\text{-}atk}}_{\mathcal{V},A} \geq {}^1\!/_2\,\mathbf{Adv}^{\mathsf{ENM\text{-}atk}}_{\mathcal{V},B}$, where $A = (A_1, A_2)$ and $B = (B_1, B_2)$ is an $\mathsf{SLNM\text{-}atk}$ and an $\mathsf{ENM\text{-}atk}$' adversary, respectively, which concludes the proof. The overall idea is that we construct the $\mathsf{SLNM\text{-}atk}$ adversary A who, provided an oracle access to the $\mathsf{ENM\text{-}atk}$' adversary B, aims to succeed in an $\mathsf{SLNM\text{-}atk}$ experiment. Note that A has to provide B with an access to $\mathsf{ENM\text{-}atk}$' oracles denoted by $\mathcal{E}_B$ and $\mathcal{D}_B^{(*)}$ while she has an access to $\mathsf{SLNM\text{-}atk}$ oracles denoted by $\mathcal{E}_A$ and $\mathcal{D}_A^{(*)}$ where $\mathcal{D}_X^{(*)}$ stands for $\mathcal{D}_X$ and $\mathcal{D}_X^{*}$, respectively. See Figure 2 for reference. Finally we show that in a half of cases, A succeeds if B succeeds (following Remark 3).

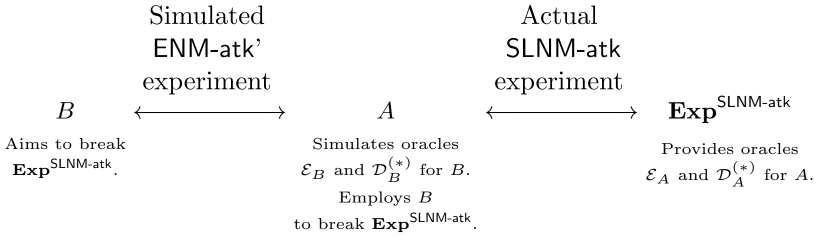
Simulated
ENM-atk'
experiment
Actual
SLNM-atk
experiment
B
A
$\mathbf{Exp}^{\mathsf{SLNM\text{-}atk}}$
Aims to break
$\mathbf{Exp}^{\mathsf{SLNM\text{-}atk}}$.
Simulates oracles
$\mathcal{E}_B$ and $\mathcal{D}_B^{(*)}$ for B.
Employs B
to break $\mathbf{Exp}^{\mathsf{SLNM\text{-}atk}}$.
Provides oracles
$\mathcal{E}_A$ and $\mathcal{D}_A^{(*)}$ for A.

Fig. 2. An overview of the interaction of adversaries A and B.

First, let us describe the simulated oracles $\mathcal{E}_B$ and $\mathcal{D}_B^{(*)}$ that employ the oracles $\mathcal{E}_A$ and $\mathcal{D}_A^{(*)}$, respectively, and share a database DB of ID-ciphertext pairs. Note that $\mathcal{E}_B$ and $\mathcal{D}_B^{(*)}$ are constructed to allow for modification of the plain data in order to succeed in particular cases that will be explained later. Nevertheless, the output b is not affected by this change since the Grant algorithm is required to output b that is independent of the input data.

```
function $\mathcal{E}_B^{(p)}(d)$
    $(b, c_b) \leftarrow \mathcal{E}_A(d + p)$
    $DB(b) \leftarrow c_b$
    return $b$
```

```
function $\mathcal{D}_B^{(*)(p)}(\mathcal{B}, \mathcal{B}_\Sigma, e)$
    if $\exists b \in \mathcal{B}_\Sigma, b \notin DB$ then return $\perp$
    for $b \in \mathcal{B}_\Sigma$ do
        $c_b \leftarrow DB(b)$
    return $d \leftarrow \mathcal{D}_A^{(*)}\big(\mathcal{B}, \bigoplus_{b\in\mathcal{B}_\Sigma} c_b \oplus \mathrm{AHE}(e)\big) -$
                $- p \cdot |\mathcal{B}_\Sigma|$
```

Next, we describe the SLNM-atk adversary A. Note that B is provided with an access to the oracles $\mathcal{E}_B$ and $\mathcal{D}_B^{(*)}$.

```
function $A_1^{\mathcal{E}_A, \mathcal{D}_A}(\mathsf{pk})$
    $p \leftarrow \{0, 1\}$
    $(M, s) \leftarrow B_1^{\mathcal{E}_B^{(p)}, \mathcal{D}_B^{(p)}}(\mathsf{pk})$
    return $\big(M, (s, p)\big)$
```

```
function $A_2^{\mathcal{E}_A, \mathcal{D}_A^*}(M, (s, p), (b^*, c^*))$
    $\big(R, (\mathcal{B}^{(i)}, \mathcal{B}_\Sigma^{(i)}, e^{(i)})_{i=1}^N\big) \leftarrow B_2^{\mathcal{E}_B^{(p)}, \mathcal{D}_B^{*(p)}}(M, s, b^*)$
    for $i = 1 \ldots N$ do
        $c^{(i)} \leftarrow \bigoplus_{b\in\mathcal{B}_\Sigma^{(i)}} c_b \oplus \mathrm{AHE}(e^{(i)})$
    if $p = 1$ then modify $R$ accordingly (described later)
    return $\big(R, (\mathcal{B}^{(i)}, c^{(i)})_{i=1}^N\big)$
```

Let us inspect the key condition of SLNM (line 7 in Definition 6) and that of ENM (line 7 in Definition 9) for $p = 0$ (the case $p = 1$ will be resolved later):

$$\forall i \in \hat{N}\colon (\mathcal{B}^{(i)}, c^{(i)}) \notin \mathcal{TB}(b^*, c^*) \wedge \perp \notin \mathbf{d} \wedge R(d_q, \mathbf{d}), \text{ and} \tag{SLNM}$$

$$\forall i \in \hat{N}\colon (e^{(i)} \neq 0 \vee \mathcal{B}^{(i)} \neq \mathcal{B}_\Sigma^{(i)}) \wedge \perp \notin \mathbf{d} \wedge R(d_q, \mathbf{d}), \tag{ENM}$$

respectively[1]. Let us focus on their relation in the scenario of this proof. By the construction of A_2, it holds $N_{\mathsf{SLNM}} = N_{\mathsf{ENM}}$ and $\mathcal{B}_{\mathsf{SLNM}}^{(i)} = \mathcal{B}_{\mathsf{ENM}}^{(i)}$. In particular, d_q is common for both experiments, let us show that also $\mathbf{d}_{\mathsf{SLNM}} = \mathbf{d}_{\mathsf{ENM}}$. We

[1] We will distinguish the variables from Equations (SLNM) and (ENM) by a subscript.

show that element-wise and omit indexes ${}^{(i)}$:

$$\begin{aligned} d_{\mathsf{SLNM}} &= \mathsf{D}_{\mathsf{sk}}(\mathcal{B}, c) = \mathsf{Dec}_{\mathsf{sk}}\big(\mathcal{B}, \mathrm{AHE}^{-1}(c)\big) = \\ &= \mathsf{Dec}_{\mathsf{sk}}\Big(\mathcal{B}, \mathrm{AHE}^{-1}\big(\bigoplus_{b \in \mathcal{B}_{\Sigma}} c_b \oplus \mathrm{AHE}(e)\big)\Big) = \\ &= \mathsf{Dec}_{\mathsf{sk}}\Big(\mathcal{B}, \sum_{b \in \mathcal{B}_{\Sigma}} \mathrm{AHE}^{-1}\big(\mathsf{E}_{\mathsf{sk}}(b, d_b)\big) + e\Big) = \\ &= \mathsf{Dec}_{\mathsf{sk}}\big(\mathcal{B}, \sum_{b \in \mathcal{B}_{\Sigma}} \mathsf{Enc}_{\mathsf{sk}}(b, d_b) + e\big) = \\ &= \mathsf{Dec}_{\mathsf{sk}}\big(\mathcal{B}, \sum_{b \in \mathcal{B}_{\Sigma}} e_b + e\big) = d_{\mathsf{ENM}} \end{aligned}$$

where d_b is a piece of data relative to b. The equations hold subsequently by:

1. line 6 in experiment $\mathbf{Exp}_{\mathcal{V},A}^{\mathsf{SLNM\text{-}atk\text{-}}q}$ in Definition 6,
2. D_{sk} in Definition 7,
3. line 4 in adversary A_2,
4. oracles $\mathcal{E}_B$ and $\mathcal{E}_A$, respectively, and the AHE homomorphic property,
5. E_{sk} in Definition 7,
6. encoding oracle $\mathcal{E}$ in Definition 9, and
7. line 6 in experiment $\mathbf{Exp}_{\mathcal{V},B}^{\mathsf{ENM\text{-}atk\text{-}}q}$ in Definition 9.

The difference between (SLNM) and (ENM) hence remains in the conditions

$$(\mathcal{B}, c) \notin \mathcal{TB}(b^*, c^*), \text{ and} \tag{21}$$

$$e \neq 0 \vee \mathcal{B} \neq \mathcal{B}_{\Sigma}. \tag{22}$$

Aiming to show $\mathbf{Adv}_{\mathcal{V},A}^{\mathsf{SLNM\text{-}atk}} \geq 1/2\, \mathbf{Adv}_{\mathcal{V},B}^{\mathsf{ENM\text{-}atk}}$ (the coefficient 1/2 will be resolved later) and following the ideas identified in Remark 3, we only need to discuss the case when a non-trivial ENM breach results in a trivial SLNM breach, i.e., when (21) does not hold while (22) holds. For the ciphertext part at the output of A_2, it holds by (3) in Definition 5 that it equals to a plain sum of ciphertexts

$$\bigoplus_{b \in \mathcal{B}_{\Sigma}} c_b \oplus \mathrm{AHE}(e) = \bigoplus_{b \in \mathcal{B}} c_b, \text{ hence} \tag{23}$$

$$e = \sum_{b \in \mathcal{B} \setminus \mathcal{B}_{\Sigma}} e_b. \tag{24}$$

In order (22) to hold, it must be $\mathcal{B} \neq \mathcal{B}_{\Sigma}$, and e encodes actual values. We distinguish two cases based on the presence of b^* in $\mathcal{B} \setminus \mathcal{B}_{\Sigma}$.

Case $b^* \in \mathcal{B} \setminus \mathcal{B}_{\Sigma}$. By (24), e shall encode a portion of the *absolutely unknown* challenge data d_1. Indeed, B only learns b^* which is independent of actual d_1. It follows that such a case may happen only accidentally, hence, by Remark 3, this is an uncontrolled type of attack which results in $\mathbf{Adv}_{\mathcal{V},B}^{\mathsf{ENM\text{-}atk}} = 0$.

Case $b^* \notin \mathcal{B} \setminus \mathcal{B}_\Sigma$**.** Here, e only encodes *known* values and poses a non-trivial valid ENM attack. However, it cannot be directly used for SLNM since it results in a trivial attack. Here comes the option $p = 1$. The idea is as follows: since B does not have any information about what data has been indeed encrypted, A can modify it before submitting it to the encryption oracle $\mathcal{E}_A$ in the construction of $\mathcal{E}_B$ oracle. When the attack is evaluated, trivial breaches are considered with respect to the *modified data* (i.e., $d + 1$), however, B returns e that encodes the original data, i.e., the difference emerges in $\mathcal{B}\setminus\mathcal{B}_\Sigma$. It follows that after respective modification, the relation R holds for the modified data while posing a non-trivial SLNM attack (modified data).

Since A_1 begins with a "coin toss" (line 2 in A_1 in this proof), the success of each branch halves, no matter which case of B_2 output occurs, hence $\mathbf{Adv}^{\mathsf{SLNM\text{-}atk}}_{\mathcal{V},A} \geq {}^1\!/_2\, \mathbf{Adv}^{\mathsf{ENM\text{-}atk}}_{\mathcal{V},B}$. □

B.2 Proof of Theorem 3

The proof follows an idea and language similar to the proof of Theorem 2 in Section B.1. In particular cf. Figure 2 since an analogous figure is omitted here.

Theorem 3 ($\mathsf{ENM\text{-}atk'} \Rightarrow \mathsf{SLNM\text{-}atk}_{\mathrm{AHE}^*}$)**.** *Let a VGE $\mathcal{V}$ use the perfect* AHE*. If its encoding is* $\mathsf{ENM\text{-}atk}$*'-secure, then it is* $\mathsf{SLNM\text{-}atk}$*-secure for* $\mathsf{CEA0}$*–*CPA*,* $\mathsf{CEA1}$*–*$\mathsf{CCA1}$ *and* $\mathsf{CEA2}$*–*$\mathsf{LCCA2}$ *pairs of* atk*'–*atk*.*

Proof. We show that $\mathbf{Adv}^{\mathsf{ENM\text{-}atk'}}_{\mathcal{V},B} \geq \mathbf{Adv}^{\mathsf{SLNM\text{-}atk(AHE^*)}}_{\mathcal{V},A}$, where $B = (B_1, B_2)$ and $A = (A_1, A_2)$ is an $\mathsf{ENM\text{-}atk}$' and an $\mathsf{SLNM\text{-}atk}_{\mathrm{AHE}^*}$ adversary, respectively, which concludes the proof. We construct the $\mathsf{ENM\text{-}atk}$' adversary B who, provided an oracle access to the $\mathsf{SLNM\text{-}atk}_{\mathrm{AHE}^*}$ adversary A, aims to succeed in an $\mathsf{ENM\text{-}atk}$' experiment. Note that B has to provide A with an access to $\mathsf{SLNM\text{-}atk}$ oracles denoted by $\mathcal{E}_A$ and $\mathcal{D}^{(*)}_A$ and to an AHE* encryption oracle $\mathcal{E}_{\mathrm{AHE}^*}$ while she has an access to $\mathsf{ENM\text{-}atk}$' oracles denoted by $\mathcal{E}_B$ and $\mathcal{D}^{(*)}_B$ where $\mathcal{D}^{(*)}_X$ stands for $\mathcal{D}_X$ and $\mathcal{D}^*_X$, respectively. Finally we show that B succeeds if A succeeds.

First, we describe the oracles $\mathcal{E}_A$ and $\mathcal{D}^{(*)}_A$, that employ the oracles $\mathcal{E}_B$ and $\mathcal{D}^{(*)}_B$, respectively, together with the $\mathcal{E}_{\mathrm{AHE}^*}$ oracle. All oracles share a database DB that implements two kinds of AHE*: an internal AHE* that serves for data encryption in $\mathcal{E}_A$ and an instance that encrypts encodings inside $\mathcal{E}_{\mathrm{AHE}^*}$. Note that these instances are distinguished in DB by 0 and 1, respectively.

```
function $\mathcal{E}_A(d)$
    $b \leftarrow \mathcal{E}_B(d)$
    $c \leftarrow \{0,1\}^\lambda$
    $DB(c) \leftarrow (b, 0)$
    return $(b, \{c\})$
```

```
function $\mathcal{E}_{\mathrm{AHE}^*}(e)$
    $c \leftarrow \{0,1\}^\lambda$
    $DB(c) \leftarrow (e, 1)$
    return $\{c\}$
```

```
function $\mathcal{D}_A^{(*)}(\mathcal{B}, \mathcal{C})$
    if $\exists c \in \mathcal{C}, c \notin DB$ then return $\perp$
    $\big(\mathcal{B}_\Sigma, (e_j, n_j)_{j=1}^n\big) \leftarrow$ decompose $\mathcal{C}$ and look up in $DB$
                                        (employ the 0 or 1 marker)
    return $d \leftarrow \mathcal{D}_B^{(*)}(\mathcal{B}, \mathcal{B}_\Sigma, \sum_{j=1}^n n_j \cdot e_j)$
```

Next, we describe the $\mathsf{ENM\text{-}atk}$' adversary B who provides A with an access to the oracles $\mathcal{E}_A$, $\mathcal{E}_{\mathrm{AHE}^*}$ and $\mathcal{D}_A^{(*)}$.

```
function $B_1^{\mathcal{E}_B, \mathcal{D}_B}(\mathsf{pk})$
    init an empty $DB$
    $(M, s) \leftarrow A_1^{\mathcal{E}_A, \mathcal{D}_A}(\mathsf{pk})$
    return $\big((M, DB), s\big)$
```

```
function $B_2^{\mathcal{E}_B, \mathcal{D}_B^*}((M, DB), s, b^*)$
    $c^* \leftarrow \{0,1\}^\lambda$
    $DB(c^*) \leftarrow (b^*, 0)$
    $\big(R, (\mathcal{B}^{(i)}, \mathcal{C}^{(i)})_{i=1}^N\big) \leftarrow A_2^{\mathcal{E}_A, \mathcal{E}_{\mathrm{AHE}^*}, \mathcal{D}_A^*}\big(M, s, (b^*, \{c^*\})\big)$
    for $i = 1 \ldots N$ do
        if $\exists c \in \mathcal{C}^{(i)}, c \notin DB$ then return $\big(R, (\emptyset, \emptyset, 0)_{i=1}^N\big)$ // something trivial
        $\big(\mathcal{B}_\Sigma^{(i)}, (e_j^{(i)}, n_j^{(i)})_{j=1}^n\big) \leftarrow$ decompose $\mathcal{C}^{(i)}$ and look up in $DB$
        $e^{(i)} \leftarrow \sum_{j=1}^n n_j^{(i)} \cdot e_j^{(i)}$
    return $\big(R, (\mathcal{B}^{(i)}, \mathcal{B}_\Sigma^{(i)}, e^{(i)})_{i=1}^N\big)$
```

The conditions of ENM (line 7 in Definition 9) and SLNM (line 7 in Definition 6) state:

$$\forall i \in \hat{N} \colon (e^{(i)} \neq 0 \vee \mathcal{B}^{(i)} \neq \mathcal{B}_\Sigma^{(i)}) \wedge \perp \notin \mathbf{d} \wedge R(d_q, \mathbf{d}), \text{ and} \qquad (\mathsf{ENM})$$

$$\forall i \in \hat{N} \colon (\mathcal{B}^{(i)}, \mathcal{C}^{(i)}) \notin \mathcal{TB}(b^*, \{c^*\}) \wedge \perp \notin \mathbf{d} \wedge R(d_q, \mathbf{d}), \qquad (\mathsf{SLNM})$$

respectively. By the construction of B_2, it holds $N_{\mathsf{ENM}} = N_{\mathsf{SLNM}}$ and $\mathcal{B}_{\mathsf{ENM}}^{(i)} = \mathcal{B}_{\mathsf{SLNM}}^{(i)}$. In particular, d_q is common for both experiments, let us show that also $\mathbf{d}_{\mathsf{ENM}} = \mathbf{d}_{\mathsf{SLNM}}$. We show that element-wise and omit indexes ${}^{(i)}$:

$$\begin{aligned} d_{\mathsf{ENM}} &= \mathsf{Dec}_{\mathsf{sk}}\big(\mathcal{B}, \sum_{b \in \mathcal{B}_\Sigma} e_b + e\big) = \\ &= \mathsf{Dec}_{\mathsf{sk}}\big(\mathcal{B}, \sum_{b \in \mathcal{B}_\Sigma} e_b + \sum_{j=1}^n n_j \cdot e_j\big) = \\ &= \mathsf{Dec}_{\mathsf{sk}}\big(\mathcal{B}, \mathrm{AHE}^{-1}(\mathcal{C})\big) = \\ &= \mathsf{D}_{\mathsf{sk}}(\mathcal{B}, \mathcal{C}) = d_{\mathsf{SLNM}}. \end{aligned}$$

The equations hold subsequently by:

1. line 6 in experiment $\mathbf{Exp}_{\mathcal{V},B}^{\mathsf{ENM\text{-}atk}\text{-}q}$ (in Definition 9),
2. adversary B_2 from this proof (line 8),
3. decomposition of $\mathcal{C}$ in adversary B_2 (line 7),
4. D_{sk} in Definition 7, and
5. line 6 in experiment $\mathbf{Exp}_{\mathcal{V},A}^{\mathsf{SLNM\text{-}atk}\text{-}q}$ (in Definition 6).

The difference between (ENM) and (SLNM) hence remains in the conditions

$$e \neq 0 \vee \mathcal{B} \neq \mathcal{B}_{\Sigma}, \text{ and} \tag{25}$$

$$(\mathcal{B},\mathcal{C}) \notin \mathcal{TB}(b^*, \{c^*\}). \tag{26}$$

Aiming to show $\mathbf{Adv}_{\mathcal{V},B}^{\mathsf{ENM\text{-}atk}'} \geq \mathbf{Adv}_{\mathcal{V},A}^{\mathsf{SLNM\text{-}atk}(\mathrm{AHE}^*)}$ and following Remark 3, we only need to discuss the case when a non-trivial SLNM breach results in a trivial ENM breach, i.e., when (25) does not hold while (26) holds. In such a case, both $e = 0$ and $\mathcal{B} = \mathcal{B}_{\Sigma}$, hence the only option for $(\mathcal{B},\mathcal{C}) \notin \mathcal{TB}(b^*, \{c^*\})$ is that $b^* \notin \mathcal{B}$, cf. (3) in Definition 5. However, such an SLNM breach is not related to the challenge data d_1 at all, hence contributes by zero to the SLNM advantage. $\mathbf{Adv}_{\mathcal{V},B}^{\mathsf{ENM\text{-}atk}'} \geq \mathbf{Adv}_{\mathcal{V},A}^{\mathsf{SLNM\text{-}atk}(\mathrm{AHE}^*)}$ follows. □

B.3 Proof of Theorem 4

The proof follows an idea and language similar to the proof of Theorem 2 in Section B.1. In particular cf. Figure 2 since an analogous figure is omitted here.

Theorem 4 ($\mathsf{INM\text{-}atk} \Rightarrow \mathsf{ENM\text{-}atk}'$)**.** *Let $\mathcal{V} = (\mathsf{Init}, \mathsf{Grant}, \mathrm{AHE}, \mathrm{RO}, \mathsf{Inc}, \mathsf{Add}, \mathsf{Idc})$ be a VGIE. If the internal encoding of $\mathcal{V}$ is INM-atk-secure, then its encoding is ENM-atk'-secure, for CIA0–CEA0, CIA1–CEA1 and CIA2–CEA2 pairs of atk–atk'.*

Proof. We show that $\mathbf{Adv}_{\mathcal{V},C}^{\mathsf{INM\text{-}atk}}(\lambda) \geq \mathbf{Adv}_{\mathcal{V},B}^{\mathsf{ENM\text{-}atk}'}(\lambda) - \mathit{negl}(\lambda)$, where $C = (C_1, C_2)$ and $B = (B_1, B_2)$ is an $\mathsf{INM\text{-}atk}$ and an $\mathsf{ENM\text{-}atk}$' adversary, respectively, which concludes the proof. We construct the $\mathsf{INM\text{-}atk}$ adversary C who, provided an oracle access to the $\mathsf{ENM\text{-}atk}$' adversary B, aims to succeed in an $\mathsf{INM\text{-}atk}$ experiment. Note that C has to provide B with an access to $\mathsf{ENM\text{-}atk}$' oracles denoted by $\mathcal{E}_B$ and $\mathcal{D}_B^{(*)}$ while she has an access to $\mathsf{INM\text{-}atk}$ oracles denoted by $\mathcal{E}_C$ and $\mathcal{D}_C^{(*)}$ where $\mathcal{D}_X^{(*)}$ stands for $\mathcal{D}_X$ and $\mathcal{D}_X^*$, respectively. Finally we show that C succeeds if B succeeds.

First, we describe the oracles $\mathcal{E}_B$ and $\mathcal{D}_B^{(*)}$, that employ the oracles $\mathcal{E}_C$ and $\mathcal{D}_C^{(*)}$, respectively.

```
function $\mathcal{E}_B(d)$
    return $b \leftarrow \mathcal{E}_C(d)$
```

```
function $\mathcal{D}_B^{(*)}(\mathcal{B}, \mathcal{B}_\Sigma, e)$
    if $\mathcal{D}_C^{(*)}(\mathcal{B}_\Sigma, e) = \bot$ then return $\bot$
    $(r, d) \leftarrow \mathcal{D}_C^{(*)}(\mathcal{B}_\Sigma, e)$
    if $r \neq \sum_{b \in \mathcal{B}} \mathrm{RO}(b)$ then return $\bot$
    return $d$
```

Next, we describe the $\mathsf{INM\text{-}atk}$' adversary C who provides B with an access to the oracles $\mathcal{E}_B$ and $\mathcal{D}_B^{(*)}$.

```
function $C_1^{\mathcal{E}_C, \mathcal{D}_C}(\mathsf{pk})$
    $(M, s) \leftarrow B_1^{\mathcal{E}_B, \mathcal{D}_B}(\mathsf{pk})$
    return $(M, s)$
```

```
function $C_2^{\mathcal{E}_C, \mathcal{D}_C^*}(M, s, b^*)$
    $\big(R, (\mathcal{B}^{(i)}, \mathcal{B}_\Sigma^{(i)}, e^{(i)})_{i=1}^N\big) \leftarrow B_2^{\mathcal{E}_B, \mathcal{D}_B^*}(M, s, b^*)$
    return $\big(R, (\mathcal{B}_\Sigma^{(i)}, e^{(i)})_{i=1}^N\big)$
```

The conditions of INM (line 7 in Definition 13) and ENM (line 7 in Definition 9) state:

$$\forall i \in \hat{N} \colon e^{(i)} \neq 0 \wedge \bot \notin (\mathbf{r}, \mathbf{d}) \wedge R(d_q, \mathbf{d}), \text{ and} \tag{INM}$$

$$\forall i \in \hat{N} \colon (e^{(i)} \neq 0 \vee \mathcal{B}^{(i)} \neq \mathcal{B}_\Sigma^{(i)}) \wedge \bot \notin \mathbf{d} \wedge R(d_q, \mathbf{d}), \tag{ENM}$$

respectively. By the construction of C_2, it holds $N_{\mathsf{INM}} = N_{\mathsf{ENM}}$ and $\mathcal{B}_{\mathsf{INM}}^{(i)} = \mathcal{B}_{\mathsf{ENM}}^{(i)}$. In particular, d_q is common for both experiments, let us show that also $\mathbf{d}_{\mathsf{INM}} = \mathbf{d}_{\mathsf{ENM}}$. We show that element-wise and omit indexes ${}^{(i)}$:

$$\begin{aligned} d_{\mathsf{INM}} &= \mathsf{Idc}_{\mathsf{sk}}\Big(\sum_{b \in \mathcal{B}_\Sigma} e_b + e\Big)[1] = \\ &= \mathsf{Dec}_{\mathsf{sk}}\Big(\mathcal{B}, \sum_{b \in \mathcal{B}_\Sigma} e_b + e\Big) = d_{\mathsf{ENM}}. \end{aligned}$$

The equations follow from line 6 in experiment $\mathbf{Exp}_{\mathcal{V},C}^{\mathsf{INM\text{-}atk\text{-}}q}$ (in Definition 13), $\mathsf{Dec}_{\mathsf{sk}}$ in Definition 12, and line 6 in experiment $\mathbf{Exp}_{\mathcal{V},B}^{\mathsf{ENM\text{-}atk\text{-}}q}$ (in Definition 9), respectively. The difference between (INM) and (ENM) hence remains in the conditions

$$e \neq 0, \text{ and} \tag{27}$$

$$e \neq 0 \vee \mathcal{B} \neq \mathcal{B}_\Sigma. \tag{28}$$

Aiming to show $\mathbf{Adv}_{\mathcal{V},C}^{\mathsf{INM\text{-}atk}}(\lambda) \geq \mathbf{Adv}_{\mathcal{V},B}^{\mathsf{ENM\text{-}atk}'}(\lambda) - negl(\lambda)$ (the negligible term $negl(\lambda)$ will be resolved later) and following Remark 3, we only need to discuss the case when a non-trivial ENM breach results in a trivial INM breach, i.e., when (27) does not hold while (28) holds. In such a case when $e = 0$ while $\mathcal{B} \neq \mathcal{B}_\Sigma$, it holds by line 5 in Definition 12, line 6 in Definition 9 and (13) in Lemma 4 that

$$r = \sum_{b\in\mathcal{B}} \mathrm{RO}(b) = \sum_{b\in\mathcal{B}_\Sigma} \mathrm{RO}(b). \tag{29}$$

Since the terms $\mathrm{RO}(b)$ are random and unknown to B, this may happen only with negligible probability. It follows $\mathbf{Adv}^{\mathsf{INM\text{-}atk}}_{\mathcal{V},C}(\lambda) \geq \mathbf{Adv}^{\mathsf{ENM\text{-}atk}}_{\mathcal{V},B}(\lambda) - negl(\lambda)$ which concludes the proof. □

B.4 Proof of Corollary 4

Corollary 4. *Assuming that* SoR *is intractable, Theorem 4 holds also for a somewhat homomorphic VGIE where the* $\mathsf{INM\text{-}atk}$ *adversary has, in addition, an access to the* RO*.*

Proof. Following the proof of Theorem 4, the difference occurs at the very end where we argue that the terms $\mathrm{RO}(b)$ are unknown to the adversary – in this case, we assume that these terms are known to the adversary. Hence, the (too large) solution $\mathbf{w^{(k)}} = (0, \ldots, 0, \frac{r_{k+1}}{GCD(r_k, r_{k+1})}, -\frac{r_k}{GCD(r_k, r_{k+1})}, 0, \ldots, 0)$ to the SoR problem identified in Remark 5 would work in a VGIE without a restriction. However and for this particular reason, we assumed a somewhat homomorphic VGIE which restricts the SoR solutions by $\|\mathbf{w}\|_1 < 2^\nu$. It follows that if (29) is satisfied for a valid VeraGreg list-ciphertext pair, it must be the case that the vector representation of $\mathcal{B} \setminus \mathcal{B}_\Sigma$ represents a vector solution to the SoR problem which we assumed to be intractable. It follows $\mathbf{Adv}^{\mathsf{INM\text{-}atk}}_{\mathcal{V},C}(\lambda) \geq \mathbf{Adv}^{\mathsf{ENM\text{-}atk}}_{\mathcal{V},B}(\lambda) - negl(\lambda)$ which concludes the proof. □

References

1. Armknecht, F., Katzenbeisser, S., Peter, A.: Group homomorphic encryption: characterizations, impossibility results, and applications. Designs, codes and cryptography **67**(2), 209–232 (2013)
2. Bellare, M., Desai, A., Pointcheval, D., Rogaway, P.: Relations among notions of security for public-key encryption schemes. In: Annual International Cryptology Conference. pp. 26–45. Springer (1998)
3. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Proceedings of the 1st ACM conference on Computer and communications security. pp. 62–73. ACM (1993)
4. Boneh, D., Segev, G., Waters, B.: Targeted malleability: homomorphic encryption for restricted computations. In: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference. pp. 350–366. ACM (2012)
5. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (leveled) fully homomorphic encryption without bootstrapping. ACM Transactions on Computation Theory (TOCT) **6**(3), 13 (2014)
6. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited. Journal of the ACM (JACM) **51**(4), 557–594 (2004)

7. Fiore, D., Gennaro, R., Pastro, V.: Efficiently verifiable computation on encrypted data. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. pp. 844–855. ACM (2014)
8. Gennaro, R., Gentry, C., Parno, B.: Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In: Annual Cryptology Conference. pp. 465–482. Springer (2010)
9. Gentry, C., Boneh, D.: A fully homomorphic encryption scheme, vol. 20. Stanford University (2009)
10. Gentry, C., Sahai, A., Waters, B.: Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In: Annual Cryptology Conference. pp. 75–92. Springer (2013)
11. Goldwasser, S., Micali, S.: Probabilistic encryption & how to play mental poker keeping secret all partial information. In: Proceedings of the fourteenth annual ACM symposium on Theory of computing. pp. 365–377. ACM (1982)
12. Halevi, S., Shoup, V.: Algorithms in HElib. In: Annual Cryptology Conference. pp. 554–571. Springer (2014)
13. Klemsa, J., Kencl, L., Vaněk, T.: VeraGreg: A Framework for Verifiable Privacy-Preserving Data Aggregation. In: 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE). pp. 1820–1825. IEEE (2018)
14. Paillier, P., et al.: Public-key cryptosystems based on composite degree residuosity classes. In: Eurocrypt. vol. 99, pp. 223–238. Springer (1999)
15. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM **21**(2), 120–126 (1978)
16. Microsoft SEAL (release 3.2). https://github.com/Microsoft/SEAL (Feb 2019), microsoft Research, Redmond, WA.