# Generic-Group Identity-Based Encryption: A Tight Impossibility Result

Gili Schul-Ganz[*] Gil Segev[*]

## Abstract

Following the pioneering work of Boneh and Franklin (CRYPTO '01), the challenge of constructing an identity-based encryption scheme based on the Diffie-Hellman assumption remained unresolved for more than 15 years. Evidence supporting this lack of success was provided by Papakonstantinou, Rackoff and Vahlis (ePrint '12), who ruled out the existence of generic-group identity-based encryption schemes supporting an identity space of sufficiently large polynomial size. Nevertheless, the breakthrough result of Döttling and Garg (CRYPTO '17) settled this long-standing challenge via a non-generic construction.

We prove a tight impossibility result for generic-group identity-based encryption, ruling out the existence of any non-trivial construction: We show that any scheme whose public parameters include $n_{\mathsf{pp}}$ group elements may support at most $n_{\mathsf{pp}}$ identities. This threshold is trivially met by any generic-group public-key encryption scheme whose public keys consist of a single group element (e.g., ElGamal encryption).

In the context of algebraic constructions, generic realizations are often both conceptually simpler and more efficient than non-generic ones. Thus, identifying exact thresholds for the limitations of generic groups is not only of theoretical significance but may in fact have practical implications when considering concrete security parameters.

[*] School of Computer Science and Engineering, Hebrew University of Jerusalem, Jerusalem 91904, Israel. Email: {gili.schul,segev}@cs.huji.ac.il. Supported by the European Union's Horizon 2020 Framework Program (H2020) via an ERC Grant (Grant No. 714253).

# 1 Introduction

Identity-based encryption [Sha84, BF01, Coc01] is one of the key pillars underlying modern cryptography, enabling a variety of access-control applications and paving a path towards more expressive forms of encryption schemes. Starting with the first realizations of identity-based encryption schemes by Boneh and Franklin [BF01] (based on the bilinear Diffie-Hellman assumption) and Cocks [Coc01] (based on the quadratic residuosity assumption) in the random-oracle model [BR93], extensive research has been devoted to constructing such schemes in the standard model (e.g., [CHK03, BB04a, BB04b, Wat05]) and based on other cryptographic assumptions (e.g., [GPV08, CHK$^+$10, ABB10]).

Despite the significant progress, a substantial gap remained for nearly two decades between the cryptographic assumptions that are known to imply public-key encryption and those that are known to imply identity-based encryption. This gap was first studied by Boneh, Papakonstantinou, Rackoff, Vahlis, and Waters [BPR$^+$08] who showed that identity-based encryption cannot be realized in a black-box manner based on trapdoor permutations or CCA-secure public-key encryption. Then, Papakonstantinou, Rackoff and Vahlis [PRV12] studied the possibility of constructing generic-group identity-based encryption schemes (i.e., identity-based encryption schemes that do not exploit any particular property of the representation of the underlying group [Sho97, Mau05]). They showed that there are no generic-group constructions of identity-based encryption schemes supporting an identity space of sufficiently large polynomial size. The result of Papakonstantinou, Rackoff and Vahlis explained, in particular, the lack of success in resolving the long-standing open problem of constructing an identity-based encryption scheme based on the Diffie-Hellman assumption. Nevertheless, the recent breakthrough of Döttling and Garg [DG17b, DG17a] settled this open problem via a non-generic construction.

**Our contribution: A tight impossibility result for generic-group IBE.** In the context of algebraic constructions, generic realizations are often both conceptually simpler and more efficient than non-generic ones. Thus, identifying exact thresholds for the limitations of generic groups is not only of theoretical significance but may in fact have practical implications when considering concrete security parameters.

For identity-based encryption schemes, such a potential threshold naturally arises by comparing the size of the scheme's identity space to the number of group elements that are included in the scheme's public parameters. Specifically, for any $n_{\mathsf{pp}} \geq 1$, already ElGamal encryption yields a generic-group identity-based encryption scheme that supports $n_{\mathsf{pp}}$ identities and whose public parameters consist of $n_{\mathsf{pp}}$ group elements (not including the group's generator). However, the work of Papakonstantinou, Rackoff and Vahlis [PRV12] only ruled out the existence of generic-group identity-based encryption schemes over an identity space of sufficiently large polynomial size[1].

We prove a tight impossibility result for constructing generic-group identity-based encryption schemes, showing that any such scheme whose public parameters consist of $n_{\mathsf{pp}}$ group elements may support up to $n_{\mathsf{pp}}$ identities. This matches the above-mentioned naive threshold that is obtained via ElGamal encryption, and more generally via any generic-group public-key encryption scheme whose public keys consist of a single group element. We prove the following theorem:

**Theorem 1.1** (Simplified). *Let $\mathcal{IBE}$ be a secure generic-group identity-based encryption scheme over an identity space $\mathcal{ID} = \{\mathcal{ID}_\lambda\}_{\lambda \in \mathbb{N}}$ whose public parameters consist of $n_{\mathsf{pp}}(\lambda)$ group elements, where $\lambda \in \mathbb{N}$ is the security parameter. Then, $|\mathcal{ID}_\lambda| \leq n_{\mathsf{pp}}(\lambda)$ for all sufficiently large $\lambda \in \mathbb{N}$.*

[1] Papakonstantinou et al. proved their result for an identity space of exponential size, but their proof seems to hold for an identity space of sufficiently large polynomial size.

We prove our result by presenting a generic-group adversary that breaks the security of any identity-based encryption scheme whose public parameters consist of $n_{\mathsf{pp}}$ group elements and supports more than $n_{\mathsf{pp}}$ identities. Our result applies to schemes satisfying a rather weak (non-adaptive) notion of security (thus ruling out, in particular, schemes that satisfy more standard notions of security), and to schemes with imperfect correctness.

Compared to the work of Papakonstantinou, Rackoff and Vahlis [PRV12], on the one hand our proof follows a similar two-step structure: We first show that any generic-group identity-based encryption scheme can be transformed into one in which secret keys do not contain group elements, and then we present an attack on any such scheme that supports more identities than the number of group elements included in its public parameters. On the other hand, however, our result does not only provide a tight impossibility result, but in fact provides a somewhat more direct technical description of our attack and of its analysis. Such a description is enabled partially due to the fact that we prove our result within Maurer's generic-group model [Mau05], whereas Papakonstantinou et al. proved their result within Shoup's incomparable generic-group model [Sho97], as discussed in Section 1.1 (e.g., in Maurer's model we do not have to take into account the additional randomness that is somewhat artificially "injected" into cryptographic computations in Shoup's model due to its random injective encoding of group elements).

Specifically, for our first step, our transformation for eliminating group elements from secret keys is essentially identical to the corresponding transformation of Papakonstantinou et al. and is provided together with a significantly more direct analysis. For our second step, our attack is based on that of Papakonstantinou et al. which relies on the common technique of attacking the security of an idealized-model scheme relative to a partly-simulated view of the model. Unlike our first step, in this step our attack and its analysis simultaneously refine and simplify those of Papakonstantinou et al. for obtaining a tight bound.

### 1.1 Overview of Our Approach

**The framework.** We prove our result within the generic-group model introduced by Maurer [Mau05], which together with the incomparable model introduced by Shoup [Sho97], seem to be the most commonly used approaches for capturing generic-group computations. At a high level, in both models algorithms have access to an oracle $\mathcal{O}$ for performing the group operation and for testing whether two group elements are equal. The difference between the two models is in the way that algorithms specify their queries to the oracle. In Maurer's model algorithms specify their queries by pointing to two group elements that have appeared in the computation so far (e.g., the 4th and the 7th group elements), whereas in Shoup's model group elements have an explicit representation (sampled uniformly at random from the set of all injective mappings from the group to sufficiently long strings) and algorithms specify their queries by providing two strings that have appeared in the computation so far as encodings of group elements.

Jager and Schwenk [JS08] proved that the complexity of any computational problem that is defined in a manner that is independent of the representation of the underlying group (e.g., computing discrete logarithms) in one model is essentially equivalent to its complexity in the other model. More generally, however, these two models are rather incomparable. On one hand, the class of cryptographic schemes that are captured by Maurer's model is a subclass of that of Shoup's model – although as demonstrated by Maurer his model still captures all schemes that only use the abstract group operation and test whether two group elements are equal. On the other hand, the same holds also for the class of adversaries, and thus in Maurer's model we have to break the security of a given scheme using an adversary that is more restricted when compared to adversaries in Shoup's model. We refer the reader to Section 2.1 for a formal description of Maurer's generic-group model.

**Generic-group identity-based encryption.** A generic-group identity-based encryption scheme $\mathcal{IBE}$ over an identity space $\mathcal{ID}$ consists of four algorithms, denoted $\mathsf{Setup}$, $\mathsf{KG}$, $\mathsf{Enc}$ and $\mathsf{Dec}$. Informally (and quite briefly), the algorithm $\mathsf{Setup}$ produces a master secret key $\mathsf{msk} \in \{0,1\}^*$ and public parameters $\mathsf{pp}$, and the algorithm $\mathsf{KG}$ on input the master secret $\mathsf{msk}$ and an identity $id \in \mathcal{ID}$ produces a secret key $\mathsf{sk}_{id}$. Next, the algorithm $\mathsf{Enc}$ on input public parameters $\mathsf{pp}$, an identity $id \in \mathcal{ID}$ and a message $b \in \{0,1\}$, produces a ciphertext $c$, which should be correctly decrypted (allowing decryption error) by the decryption algorithm $\mathsf{Dec}$ using the secret key $\mathsf{sk}_{id}$. The outputs of these four algorithms may consist of a combination of group elements and an explicit string, with the exception of assuming without loss of generality that the master secret key $\mathsf{msk}$ is always an explicit string (e.g., the internal randomness on which $\mathsf{Setup}$ is invoked).

**The structure of our proof.** We prove our result by presenting a generic-group adversary that breaks the security of any identity-based encryption scheme whose public parameters $\mathsf{pp}$ consist of $n_{\mathsf{pp}}$ group elements (and, possibly, an additional explicit string) and supports more than $n_{\mathsf{pp}}$ identities. As mentioned above, at a high level, we follow a two-step structure similar to that introduced in the work of Papakonstantinou et al. [PRV12]: We first show that any generic-group identity-based encryption scheme can be transformed into one in which secret keys do not contain group elements (while modifying only its key-generation and decryption algorithms), and then we present an attack on any such scheme that supports more identities than the number of group elements included in its public parameters. The remainder of this section consists of a high-level informal description of these two steps (we note that the following description omits crucial technical details, and we refer the reader to the relevant sections for formal descriptions and proofs).

In what follows, given a generic-group identity-based encryption scheme we let $pp_1, \ldots, pp_{n_{\mathsf{pp}}}$, $sk_{id,1}, \ldots, sk_{id,n_{\mathsf{sk}}}$ and $c_1, \ldots, c_{n_{\mathsf{ct}}}$ denote the group elements included in its public parameters $\mathsf{pp}$ and in each of its secret keys $\mathsf{sk}_{id}$ and ciphertexts $c$, respectively (for simplicity, we assume throughout this informal overview that public parameters, secret keys and ciphertexts do not additionally contain explicit strings).

**Step I: Eliminating group elements from secret keys.** Given a generic-group identity-based encryption scheme $\mathcal{IBE} = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$, we modify its key-generation algorithm $\mathsf{KG}$ and decryption algorithm $\mathsf{Dec}$ as follows:

- The modified key-generation algorithm $\widetilde{\mathsf{KG}}$ on input the public parameters $\mathsf{pp}$, the master secret key $\mathsf{msk} \in \{0,1\}^*$ and an identity $id \in \mathcal{ID}$, first produces a secret key $\mathsf{sk}_{id}$ by invoking the underlying key-generation algorithm $\mathsf{KG}$. Then, for each message $b \in \{0,1\}$, it repeatedly computes $\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_{id}, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, id, b))$ using fresh randomness for $\mathsf{Enc}$ and $\mathsf{Dec}$, and collects into a set $\mathcal{L}_{id}$ all linear equations that result from the positively-answered equality queries in these computations. Note that since the group elements that are given as input to these computations are those included in $\mathsf{pp}$ and $\mathsf{sk}_{id}$ (as well as the generator $1 \in \mathbb{Z}_N$ that is given as input to all computations), then each such equation is of the form $\alpha_0 \cdot 1 + \sum_{\ell=1}^{n_{\mathsf{pp}}} \alpha_\ell \cdot pp_\ell + \sum_{\ell=1}^{n_{\mathsf{sk}}} \beta_\ell \cdot sk_{id,\ell} = 0$ for some coefficients $\alpha_0, \ldots, \alpha_{n_{\mathsf{pp}}}, \beta_1, \ldots, \beta_{n_{\mathsf{sk}}} \in \mathbb{Z}_N$. The algorithm then outputs the modified secret key $\widetilde{\mathsf{sk}_{id}} = \mathcal{L}_{id}$ which consists of $(n_{\mathsf{pp}} + n_{\mathsf{sk}} + 1)$-dimensional vectors of coefficients over $\mathbb{Z}_N$ (and does not contain group elements).
- The modified decryption algorithm $\widetilde{\mathsf{Dec}}$ on input the public parameters $\mathsf{pp}$, a modified secret key $\widetilde{\mathsf{sk}_{id}} = \mathcal{L}_{id}$ and a ciphertext $c$, emulates the computation $\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_{id}, c)$ using symbolic variables instead of the group elements $sk_{id,1}, \ldots, sk_{id,n_{\mathsf{sk}}}$ included in the secret key $\mathsf{sk}_{id}$. As

long as it is able to obtain and to respond with the correct answer to all emulated equality queries, then the emulation will be identical to the actual computation $\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_{id}, c)$.

Note that since the group elements that are given as input to the actual computation are those included in $\mathsf{pp}$, $\mathsf{sk}_{id}$ and $c$ (as well as the generator $1 \in \mathbb{Z}_N$), then each emulated equality query corresponds to a linear equation of the form $\alpha_0 \cdot 1 + \sum_{\ell=1}^{n_{\mathsf{pp}}} \alpha_\ell \cdot pp_\ell + \sum_{\ell=1}^{n_{\mathsf{sk}}} \beta_\ell \cdot sk_{id,\ell} + \sum_{\ell=1}^{n_{\mathsf{ct}}} \gamma_i \cdot c_i = 0$, for coefficients $\alpha_0, \ldots, \alpha_{n_{\mathsf{pp}}}, \beta_1, \ldots, \beta_{n_{\mathsf{sk}}}, \gamma_1, \ldots, \gamma_{n_{\mathsf{ct}}} \in \mathbb{Z}_N$. Now, the algorithm $\widetilde{\mathsf{Dec}}$ uses the set $\mathcal{L}_{id}$ and the actual oracle $\mathcal{O}$ for responding to this query as follows. If there exist $\alpha'_0, \ldots, \alpha'_{n_{\mathsf{pp}}} \in \mathbb{Z}_N$ such that $(\alpha'_0, \ldots, \alpha'_{n_{\mathsf{pp}}}, \beta_1, \ldots, \beta_{n_{\mathsf{sk}}}) \in \mathsf{span}(\mathcal{L}_{id})$, then $\widetilde{\mathsf{Dec}}$ issues to the actual oracle $\mathcal{O}$ an equality queries corresponding to the linear equation $(\alpha_0 - \alpha'_0) \cdot 1 + \sum_{\ell=1}^{n_{\mathsf{pp}}} (\alpha_\ell - \alpha'_\ell) \cdot pp_\ell + \sum_{\ell=1}^{n_{\mathsf{ct}}} \gamma_\ell \cdot c_\ell = 0$, and return its output as the response. If there do not exist such $\alpha'_0, \ldots, \alpha'_{n_{\mathsf{pp}}} \in \mathbb{Z}_N$, then $\widetilde{\mathsf{Dec}}$ responds negatively.

In other words, the algorithm $\widetilde{\mathsf{Dec}}$ uses the knowledge provided by the set $\mathcal{L}_{id}$ in order to translate each equality query involving the group elements of $\mathsf{pp}$, $\mathsf{sk}_{id}$ and $c$ into an equality queries that involves the group elements of only $\mathsf{pp}$ and $c$. A simple probabilistic argument (see Claim 3.2) shows that this translation introduces only an arbitrary polynomially-small decryption error $1/p(\lambda)$ when setting the number of iterations performed by the modified key-generation algorithm to $p(\lambda) \cdot (n_{\mathsf{pp}}(\lambda) + n_{\mathsf{sk}}(\lambda))$.

**Step II: Our attack.** Let $\mathcal{IBE} = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ be a generic-group identity-based encryption scheme over an identity space $\mathcal{ID}$ whose public parameters consist of $n_{\mathsf{pp}}$ group elements, whose secret keys do not contain group elements, and that supports at least $n_{\mathsf{pp}}+1$ identities. For simplicity and without loss of generality we assume that $\{1, \ldots, n_{\mathsf{pp}} + 1\} \subseteq \mathcal{ID}$.

The key observation underlying our attack is based on considering the set of linear equations that result from the positively-answered equality queries in the computations $\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_{id}, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, id, b))$ for each message $b \in \{0, 1\}$ and identity $id \in \{1, \ldots, n_{\mathsf{pp}} + 1\}$. Given that the secret keys $\mathsf{sk}_{id}$ do not contain any group elements, then the group elements that are given as input to these computations are only those that are included in the public parameters $\mathsf{pp}$ (as well as the generator $1 \in \mathbb{Z}_N$ that is given as input to all computations). Thus, each such equation is of the form $\alpha_0 \cdot 1 + \sum_{\ell=1}^{n_{\mathsf{pp}}} \alpha_\ell \cdot pp_\ell = 0$ for some coefficients $\alpha_0, \ldots, \alpha_{n_{\mathsf{pp}}} \in \mathbb{Z}_N$. Given that $(1, \mathsf{pp}_1, \ldots, \mathsf{pp}_{n_{\mathsf{pp}}})$ is a non-zero vector, then the vectors of coefficients of these sets of equations span a linear subspace of dimension at most $n_{\mathsf{pp}}$.

Therefore, for at least one identity $id \in \{1, \ldots, n_{\mathsf{pp}} + 1\}$, it must be the case that the set of linear equations that result from the positively-answered equality queries in the computation $\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_{id}, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, id, b))$ is contained in the linear subspace spanned by the sets of linear equations that result from the positively-answered equality queries in the computations

$$\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_1, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, 1, b)), \ldots, \mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_{id-1}, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, id - 1, b)).$$

Moreover, once our adversary discovers this subspace by using the secret keys $\mathsf{sk}_1, \ldots, \mathsf{sk}_{id-1}$, then it can intuitively generate alternative public parameters $\mathsf{pp}^*$ that are consistent with this subspace, together with a matching alternative master secret key $\mathsf{msk}^*$. Then, it uses the alternative public parameters $\mathsf{pp}^*$ and master secret key $\mathsf{msk}^*$ for generating an alternative secret key $\mathsf{sk}^*_{id}$ for decrypting the challenge ciphertext. The correctness of the scheme guarantees that, with high probability, $\mathsf{sk}^*_{id}$ will decrypt correctly a ciphertext that is encrypted and decrypted relative to $\mathsf{pp}^*$, and we show that $\mathsf{sk}^*_{id}$ is in fact useful also when encrypting and decrypting relative to $\mathsf{pp}$.

### 1.2 Paper Organization

The remainder of this paper is organized as follows. First, in Section 2 we present the basic notation used throughout the paper, and formally describe the framework of generic-group identity-based encryption. Then, in Section 3 we show that any generic-group identity-based encryption scheme can be transformed into one in which secret keys do not contain group elements. Finally, in Section 4 we present an attack on any generic-group identity-based encryption scheme whose secret keys do not contain group elements, and that supports more identities than the number of group elements included in its public parameters.

## 2 Preliminaries

In this section we present the basic notions and standard cryptographic tools that are used in this work. For a distribution $X$ we denote by $x \leftarrow X$ the process of sampling a value $x$ from the distribution $X$. Similarly, for a set $\mathcal{X}$ we denote by $x \leftarrow \mathcal{X}$ the process of sampling a value $x$ from the uniform distribution over $\mathcal{X}$. For an integer $n \in \mathbb{N}$ we denote by $[n]$ the set $\{1, \ldots, n\}$. A function $\nu : \mathbb{N} \to \mathbb{R}^+$ is *negligible* if for any polynomial $p(\cdot)$ there exists an integer $N$ such that for all $n > N$ it holds that $\nu(n) \leq 1/p(n)$.

### 2.1 Generic Groups and Algorithms

We prove our results within the generic-group model introduced by Maurer [Mau05]. We consider computations in cyclic groups of order $N$ (all of which are isomorphic to $\mathbb{Z}_N$ with respect to addition modulo $N$), for a $\lambda$-bit prime $N$ that is generated by an order-generation algorithm $\mathsf{PrimeGen}(1^\lambda)$, where $\lambda \in \mathbb{N}$ is the security parameter.

When considering such groups, each computation in Maurer's model is associated with a table $\mathbf{B}$. Each entry of this table stores an element of $\mathbb{Z}_N$, and we denote by $V_i$ the group element that is stored in the $i$th entry. Generic algorithms access this table via an oracle $\mathcal{O}$, providing black-box access to $\mathbf{B}$ as follows. A generic algorithm $\mathcal{A}$ that takes $d$ group elements as input (along with an optional bit-string) does not receive an explicit representation of these group elements, but instead, has oracle access to the table $\mathbf{B}$, whose first $d$ entries store the $\mathbb{Z}_N$ elements corresponding to the $d$ group elements in $\mathcal{A}$'s input. That is, if the input of an algorithm $\mathcal{A}$ is a tuple $(g_1, \ldots, g_d, x)$, where $g_1, \ldots, g_d$ are group elements and $x$ is an arbitrary string, then from $\mathcal{A}$'s point of view the input is the tuple $(\widehat{g_1}, \ldots, \widehat{g_d}, x)$, where $\widehat{g_1}, \ldots, \widehat{g_d}$ are pointers to the group elements $g_1, \ldots, g_d$ (these group elements are stored in the table $\mathbf{B}$), and $x$ is given explicitly.

All generic algorithms in this paper receive as input the order $N$ and a generator of the group (we capture this fact by always assuming that the first entry of $\mathbf{B}$ is occupied by $1 \in \mathbb{Z}_N$). The oracle $\mathcal{O}$ allows for two types of queries:

- **Group-operation queries:** On input $(i, j, \circ)$ for $i, j \in \mathbb{N}$ and $\circ \in \{+, -\}$, the oracle checks that the $i$th and $j$th entries of the table $\mathbf{B}$ are not empty, computes $V_i \circ V_j \bmod N$ and stores the result in the next available entry. If either the $i$th or the $j$th entries are empty, the oracle ignores the query.
- **Equality queries:** On input $(i, j, =)$ for $i, j \in \mathbb{N}$, the oracle checks that the $i$th and $j$th entries of the table $\mathbf{B}$ are not empty, and then returns 1 if $V_i = V_j$ and 0 otherwise. If either the $i$th or the $j$th entries are empty, the oracle ignores the query.

In this paper we consider interactive computations in which multiple algorithms pass group elements (as well as non-group elements) as inputs to one another. This is naturally supported by the model as follows: When a generic algorithm $\mathcal{A}$ outputs $k$ group elements (along with a potential bit-string $\sigma$), it outputs the indices of $k$ (non-empty) entries in the table $\mathbf{B}$ (together with $\sigma$). When these outputs (or some of them) are passed on as inputs to a generic algorithm $\mathcal{C}$, the table $\mathbf{B}$ is re-initialized, and these values (and possibly additional group elements that $\mathcal{C}$ receives as input) are placed in the first entries of the table. Additionally, we rely on the following conventions:

1. Throughout the paper we refer to values as either "explicit" ones or "implicit" ones. Explicit values are all values whose representation (e.g., binary strings of a certain length) is explicitly provided to the generic algorithms under consideration. Implicit values are all values that correspond to group elements and that are stored in the table $\mathbf{B}$ – thus generic algorithms can access them only via oracle queries. We will sometimes interchange between providing group elements as input to generic algorithms implicitly, and providing them explicitly. Note that moving from the former to the latter is well defined, since a generic algorithm $\mathcal{A}$ that receives some of its input group elements explicitly can always simulate the computation as if they were received as part of the table $\mathbf{B}$.
2. For a group element $g$, we will differentiate between the case where $g$ is provided explicitly and the case where it is provided implicitly via the table $\mathbf{B}$, using the notation $g$ in the former case, and the notation $\widehat{g}$ in the latter. Additionally, we extend this notation to a vector $v$ of group elements, which may be provided either explicitly (denoted $v$) or implicitly via the table $\mathbf{B}$ (denoted $\widehat{v}$).

### 2.2 Generic-Group Identity-Based Encryption

The following definition adapts the standard notion of an identity-based encryption scheme to the generic-group model.

**Definition 2.1.** A generic-group identity-based encryption scheme over an identity space $\mathcal{ID} = \{\mathcal{ID}_\lambda\}_{\lambda\in\mathbb{N}}$ is a quadruple $\mathcal{IBE} = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ of generic algorithms defined as follows:

- The algorithm $\mathsf{Setup}$ is a probabilistic algorithm that receives as input the security parameter $\lambda \in \mathbb{N}$ and the group order $N$, and outputs a master secret key $\mathsf{msk} \in \{0,1\}^*$ and public parameters $\mathsf{pp} = (\mathsf{pp}_\mathsf{G}, \mathsf{pp}_\mathsf{str})$, where $\mathsf{pp}_\mathsf{G}$ is a tuple of $n_\mathsf{pp}$ group elements and $\mathsf{pp}_\mathsf{str}$ is a binary string.
- The algorithm $\mathsf{KG}$ is a (potentially) probabilistic algorithm that receives as input public parameters $\mathsf{pp}$, a master secret key $\mathsf{msk}$ and an identity $id$. It outputs an identity secret key $\mathsf{sk}_{id} = (\mathsf{sk}_{id,\mathsf{G}}, \mathsf{sk}_{id,\mathsf{str}})$, where $\mathsf{sk}_{id,\mathsf{G}}$ is a tuple of group elements and $\mathsf{sk}_{id,\mathsf{str}}$ is a binary string.
- The algorithm $\mathsf{Enc}$ is a probabilistic algorithm that receives as input public parameters $\mathsf{pp}$, an identity $id$, and a bit $b \in \{0,1\}$. It outputs a ciphertext $c = (c_\mathsf{G}, c_\mathsf{str})$, where $c_\mathsf{G}$ is a tuple of group elements and $c_\mathsf{str}$ is a binary string.
- The algorithm $\mathsf{Dec}$ is a (potentially) probabilistic algorithm that receives as input public parameters $\mathsf{pp}$, an identity secret key $\mathsf{sk}_{id}$, and a ciphertext $c$. It outputs either a bit $b \in \{0,1\}$ or the special rejection symbol $\perp$.

We consider the standard correctness and security requirements of identity-based encryption schemes. In fact, we consider a rather weak notion of non-adaptive security asking the attacker to choose both the challenge identity and the identities for which secret keys are provided ahead of time (since we prove an impossibility result then this can only strengthen our result).

**Definition 2.2.** A generic-group identity-based encryption scheme $\mathcal{IBE} = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ over an identity space $\mathcal{ID} = \{\mathcal{ID}_\lambda\}_{\lambda\in\mathbb{N}}$ has decryption error $\epsilon = \epsilon(\lambda)$ if for any security parameter

$\lambda \in \mathbb{N}$, for any $N$ produced by $\mathsf{PrimeGen}(1^\lambda)$, for any $(\mathsf{msk}, \mathsf{pp})$ produced by $\mathsf{Setup}^{\mathcal{O}}(1^\lambda)$, for any $id \in \mathcal{ID}_\lambda$, and for any $b \in \{0,1\}$ it holds that

$$\Pr\left[\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_{id}, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, id, b)) = b\right] \geq 1 - \epsilon$$

where $\mathsf{sk}_{id} \leftarrow \mathsf{KG}^{\mathcal{O}}(\mathsf{pp}, \mathsf{msk}, id)$, and the probability is taken over the internal randomness of the algorithms $\mathsf{KG}$, $\mathsf{Enc}$ and $\mathsf{Dec}$.

We note that our results can be easily adapted to a more relaxed notion of correctness, asking that the above holds for almost all $(\mathsf{msk}, \mathsf{pp})$ produced by $\mathsf{Setup}^{\mathcal{O}}(1^\lambda)$ instead of for all such $(\mathsf{msk}, \mathsf{pp})$.

**Definition 2.3.** A generic-group identity-based encryption scheme $\mathcal{IBE} = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ over an identity space $\mathcal{ID} = \{\mathcal{ID}_\lambda\}_{\lambda \in \mathbb{N}}$ is non-adaptively secure if for any generic-group algorithm $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ that issues a polynomial number of queries there exists a negligible function $\nu(\lambda)$ such that

$$\left| \Pr\left[\mathsf{Expt}_{\mathcal{IBE},\mathcal{A}}(\lambda) = 1\right] - \frac{1}{2} \right| \leq \nu(\lambda)$$

for all sufficiently large $\lambda \in \mathbb{N}$, where the experiment $\mathsf{Expt}_{\mathcal{IBE},\mathcal{A}}(\lambda)$ is defined as follows:

1. $N \leftarrow \mathsf{PrimeGen}(1^\lambda)$.
2. $(id^*, id_1, \ldots id_k, \mathsf{state}) \leftarrow \mathcal{A}_1^{\mathcal{O}}(1^\lambda, N)$, for a polynomial $k = k(\lambda)$, where $id^*, id_1, \ldots id_k \in \mathcal{ID}_\lambda$ and $id^* \notin \{id_1, \ldots id_k\}$.
3. $(\mathsf{msk}, \mathsf{pp}) \leftarrow \mathsf{Setup}^{\mathcal{O}}(1^\lambda, N)$.
4. $\mathsf{sk}_{id_i} \leftarrow \mathsf{KG}^{\mathcal{O}}(\mathsf{pp}, \mathsf{msk}, id_i)$ for $i \in [k]$.
5. $c^* \leftarrow \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, id^*, b)$ for $b \leftarrow \{0,1\}$.
6. $b' \leftarrow \mathcal{A}_2^{\mathcal{O}}(\mathsf{state}, \mathsf{pp}, c^*, \mathsf{sk}_{id_1}, \ldots, \mathsf{sk}_{id_k})$.
7. If $b' = b$ then output 1, and otherwise output 0.

## 3 Eliminating Group Elements From Secret Keys

In this section we show that any generic-group identity-based encryption scheme can be transformed into one in which secret keys do not contain group elements. The transformation supports the same identity space, and does not modify the scheme's setup and encryption procedures (in particular, it does not increase the number of group elements that are contained in the scheme's public parameters). The transformation does modify the scheme's key-generation and decryption algorithms, leading to an arbitrary polynomially-small increase in the scheme's decryption error. We prove the following theorem:

**Theorem 3.1.** *Let $\mathcal{IBE}$ be a generic-group identity-based encryption scheme over an identity space $\mathcal{ID} = \{\mathcal{ID}_\lambda\}_{\lambda \in \mathbb{N}}$ with decryption error $\epsilon(\lambda)$ and whose public parameters consist of $n_{\mathsf{pp}}(\lambda)$ group elements. Then, for any polynomial $p(\lambda)$, there exists a generic-group identity-based encryption scheme $\widetilde{\mathcal{IBE}}$ over the identity space $\mathcal{ID}$ with decryption error $\epsilon(\lambda) + 1/p(\lambda)$, whose public parameters consist of $n_{\mathsf{pp}}(\lambda)$ group elements, and whose secret keys do not contain group elements.*

**Preliminaries.** Let $A$ be a generic-group algorithm that receives as input group elements $g_1, \ldots, g_k$ (in addition to the group element $1 \in \mathbb{Z}_N$ that is always provided as the first input to all algorithms) and an explicit string $\mathsf{str}$. We let $\mathcal{EQ}\left(A^{\mathcal{O}}(\widehat{g_1}, \ldots, \widehat{g_k}, \mathsf{str})\right)$ denote the random variable corresponding to the set of all $(k+1)$-dimensional vectors over $\mathbb{Z}_N$ resulting from the positively-answered equality queries in the (possibly randomized) computation $A^{\mathcal{O}}(\widehat{g_1}, \ldots, \widehat{g_k}, \mathsf{str})$.

Formally, for each equality query $(i, j)$ that is positively answered during this computation, let $V_i$ and $V_j$ denote the group elements that are located in the $i$th and $j$th entries of the table $\mathbf{B}$ associated with oracle $\mathcal{O}$ in this computation (i.e., $V_i$ and $V_j$ are the two group elements for which $A$ issues this equality query). Then, $V_i$ and $V_j$ are linear combinations of the group elements $1, g_1, \ldots, g_k$ that are provided as input to the computation, and the coefficients of these linear combinations can be determined by keeping track of the computation's group-operation queries. Let $V_i - V_j = \alpha_0 \cdot 1 + \sum_{\ell=1}^{k} \alpha_\ell \cdot g_\ell$ for $\alpha_0, \ldots, \alpha_k \in \mathbb{Z}_N$. The set $\mathcal{EQ}\left(A^{\mathcal{O}}(\widehat{g_1}, \ldots, \widehat{g_k}, \mathsf{str})\right)$ consists of all such vectors $(\alpha_0, \ldots, \alpha_k) \in \mathbb{Z}_N^{k+1}$.

In addition, for a generic-group identity-based encryption $\mathcal{IBE} = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$, and for any public parameters $\mathsf{pp}$ produced by $\mathsf{Setup}$ we let $\mathsf{pp} = (pp_1, \ldots, pp_{n_{\mathsf{pp}}}, \mathsf{pp}_{\mathsf{str}})$, where $pp_1, \ldots, pp_{n_{\mathsf{pp}}}$ are group elements and $\mathsf{pp}_{\mathsf{str}}$ is an explicit string (recall that any $\mathsf{msk}$ produced by $\mathsf{Setup}$ is an explicit string). Similarly, for any secret key $\mathsf{sk}_{id}$ produced by $\mathsf{KG}$ we let $\mathsf{sk}_{id} = (sk_{id,1}, \ldots, sk_{id,n_{\mathsf{sk}}}, \mathsf{sk}_{id,\mathsf{str}})$ where $sk_{id,1}, \ldots, sk_{id,n_{\mathsf{sk}}}$ are group elements and $\mathsf{sk}_{id,\mathsf{str}}$ is an explicit string, and for any ciphertext $c$ produced by $\mathsf{Enc}$ we let $c = (c_1, \ldots, c_{n_{\mathsf{ct}}}, c_{\mathsf{str}})$, where $c_1, \ldots, c_{n_{\mathsf{ct}}}$ are group elements and $c_{\mathsf{str}}$ is an explicit string.

Finally, our proof relies on the following lemma (which is proved in Appendix A):

**Lemma 3.2.** *Let $k \geq 1$, and let $X_1, \ldots, X_k$ be independent and identically distributed random variables over subsets of a linear vector space $V$ of dimension $\mathsf{dim}(V)$. Then,*

$$\Pr\left[X_k \not\subseteq \mathsf{span}\left(X_1 \cup \cdots \cup X_{k-1}\right)\right] \leq \frac{\mathsf{dim}(V)}{k}.$$

The remainder of this section consists of the proof of Theorem 3.1.

**Proof of Theorem 3.1.** Let $\mathcal{IBE} = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$, and let $p = p(\lambda)$ be a polynomial. We construct a generic-group identity-based encryption scheme $\widetilde{\mathcal{IBE}} = (\widetilde{\mathsf{Setup}}, \widetilde{\mathsf{KG}}, \widetilde{\mathsf{Enc}}, \widetilde{\mathsf{Dec}})$ by letting $\widetilde{\mathsf{Setup}} = \mathsf{Setup}$ and $\widetilde{\mathsf{Enc}} = \mathsf{Enc}$, and by defining the algorithms $\widetilde{\mathsf{KG}}$ and $\widetilde{\mathsf{Dec}}$ as follows.

**The key-generation algorithm $\widetilde{\mathsf{KG}}^{\mathcal{O}}(\mathsf{pp}, \mathsf{msk}, id)$:**

1. Generate $\mathsf{sk}_{id} = (\widehat{sk_{id,1}}, \ldots, \widehat{sk_{id,n_{\mathsf{sk}}}}, \mathsf{sk}_{id,\mathsf{str}}) \leftarrow \mathsf{KG}^{\mathcal{O}}(\mathsf{pp}, \mathsf{msk}, id)$.
2. For every $b \in \{0,1\}$ and $i \in [M-1]$, where $M = p \cdot (n_{\mathsf{pp}} + n_{\mathsf{sk}})$, compute $\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_{id}, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, id, b))$ using fresh randomness for $\mathsf{Enc}$ and $\mathsf{Dec}$, and let
$$\mathcal{L}_{id,b,i} = \mathcal{EQ}\left(\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_{id}, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, id, b))\right) \subseteq \mathbb{Z}_N^{n_{\mathsf{pp}}+n_{\mathsf{sk}}+1}.$$
3. Output $\widetilde{\mathsf{sk}_{id}} = (\mathcal{L}_{id}, \mathsf{sk}_{id,\mathsf{str}})$, where $\mathcal{L}_{id} = \bigcup_{b\in\{0,1\}, i\in[M-1]} \mathcal{L}_{id,b,i}$.

**The decryption algorithm $\widetilde{\mathsf{Dec}}^{\mathcal{O}}(\mathsf{pp}, \widetilde{\mathsf{sk}_{id}}, c)$:**

1. Let $\mathsf{pp} = (\widehat{pp_1}, \ldots, \widehat{pp_{n_{\mathsf{pp}}}}, \mathsf{pp}_{\mathsf{str}})$, $\widetilde{\mathsf{sk}_{id}} = (\mathcal{L}_{id}, \mathsf{sk}_{id,\mathsf{str}})$, and $c = (\widehat{c_1}, \ldots, \widehat{c_{n_{\mathsf{ct}}}}, c_{\mathsf{str}})$.
2. Emulate the computation $\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_{id}, c)$ using symbolic variables instead of $sk_{id,1}, \ldots, sk_{id,n_{\mathsf{sk}}}$ (recall that $\mathsf{sk}_{id} = (\widehat{sk_{id,1}}, \ldots, \widehat{sk_{id,n_{\mathsf{sk}}}}, \mathsf{sk}_{id,\mathsf{str}})$) by responding to each equality query $(i, j)$ as follows:

(a) Let $V_i$ and $V_j$ denote the corresponding group elements, and let

$$V_i - V_j = \alpha_0 \cdot 1 + \sum_{\ell=1}^{n_{\mathsf{pp}}} \alpha_\ell \cdot pp_\ell + \sum_{\ell=1}^{n_{\mathsf{sk}}} \beta_\ell \cdot sk_{id,\ell} + \sum_{\ell=1}^{n_{\mathsf{ct}}} \gamma_i \cdot c_i,$$

where $\alpha_0, \ldots, \alpha_{n_{\mathsf{pp}}}, \beta_1, \ldots, \beta_{n_{\mathsf{sk}}}, \gamma_1, \ldots, \gamma_{n_{\mathsf{ct}}} \in \mathbb{Z}_N$ (as explained above, these coefficients can be found by keeping track of the emulated computation's group-operation queries ).

(b) If there exist $\alpha'_0, \ldots, \alpha'_{n_{\mathsf{pp}}} \in \mathbb{Z}_N$ such that $(\alpha'_0, \ldots, \alpha'_{n_{\mathsf{pp}}}, \beta_1, \ldots, \beta_{n_{\mathsf{sk}}}) \in \mathsf{span}(\mathcal{L}_{id})$, then issue group-operation queries for positioning the group element $W_{i,j} = (\alpha_0 - \alpha'_0) \cdot 1 + \sum_{\ell=1}^{n_{\mathsf{pp}}} (\alpha_\ell - \alpha'_\ell) \cdot pp_\ell + \sum_{\ell=1}^{n_{\mathsf{ct}}} \gamma_\ell \cdot c_\ell$ in the table $\mathbf{B}$. If $W_{i,j} = 0$ (this can be determined by issuing a single equality query), then answer the equality query $(i,j)$ positively, and otherwise answer it negatively.

(c) If there do not exist such $\alpha'_0, \ldots, \alpha'_{n_{\mathsf{pp}}} \in \mathbb{Z}_N$, then answer the equality query $(i,j)$ negatively.

3. Output the result of the emulated computation.

First, in terms of efficiency, note that if the algorithms $(\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ issue at most a polynomial number of queries, then so do the algorithms $(\widetilde{\mathsf{Setup}}, \widetilde{\mathsf{KG}}, \widetilde{\mathsf{Enc}}, \widetilde{\mathsf{Dec}})$. Second, in terms of security, note that the scheme $\widetilde{\mathcal{IBE}}$ is at least as secure as the scheme $\mathcal{IBE}$: The schemes have the same setup and encryption algorithms, and the modified key-generation algorithm $\widetilde{\mathsf{KG}}$ is defined by applying a poly-query procedure to the output of the underlying key-generation algorithm $\mathsf{KG}$. Therefore, any adversary attacking the scheme $\widetilde{\mathcal{IBE}}$ while issuing a polynomial number of queries (recall Definition 2.3) can be efficiently transformed into an adversary attacking the scheme $\mathcal{IBE}$ while issuing a polynomial number of queries and with (at least) the same advantage.

We are thus left with bounding the decryption error of the scheme $\widetilde{\mathcal{IBE}}$ (recall Definition 2.2). Fix a security parameter $\lambda \in \mathbb{N}$, an integer $N$ that is produced by $\mathsf{PrimeGen}(1^\lambda)$, a pair $(\mathsf{msk}, \mathsf{pp})$ that is produced by $\mathsf{Setup}^{\mathcal{O}}(1^\lambda)$, an identity $id \in \mathcal{ID}_\lambda$, and a message $b \in \{0,1\}$. The scheme $\mathcal{IBE}$ has decryption error at most $\epsilon(\lambda)$, and therefore

$$\begin{aligned} \Pr&\left[\widetilde{\mathsf{Dec}}^{\mathcal{O}}(\mathsf{pp}, \widetilde{\mathsf{sk}_{id}}, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, id, b)) \neq b\right] \\ &\leq \epsilon(\lambda) + \Pr\left[\widetilde{\mathsf{Dec}}^{\mathcal{O}}(\mathsf{pp}, \widetilde{\mathsf{sk}_{id}}, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, id, b)) \neq \mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_{id}, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, id, b))\right] \end{aligned} \tag{3.1}$$

In order to bound the probability in which the computations $\widetilde{\mathsf{Dec}}^{\mathcal{O}}(\mathsf{pp}, \widetilde{\mathsf{sk}_{id}}, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, id, b))$ and $\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_{id}, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, id, b))$ do not produce the same output, it suffices to bound the probability in which an equality query is answered positively in one computation but negatively in the other computation (as long as the responses to all equality queries are consistent then the emulated computation carried out by $\widetilde{\mathsf{Dec}}$ perfectly simulates $\mathsf{Dec}$'s computation).

Assuming that the responses to equality queries are consistent among the two computations up to a certain point, then both computations issue the exact same next equality query $(i,j)$. Following the description of $\widetilde{\mathsf{Dec}}$, let $V_i$ and $V_j$ denote the group elements in the $i$th and $j$th entries of the emulated table $\widetilde{\mathbf{B}}$, and let $V_i - V_j = \alpha_0^{(t)} \cdot 1 + \sum_{\ell=1}^{n_{\mathsf{pp}}} \alpha_\ell^{(t)} \cdot pp_\ell + \sum_{\ell=1}^{n_{\mathsf{sk}}} \beta_\ell^{(t)} \cdot sk_{id,\ell} + \sum_{\ell=1}^{n_{\mathsf{ct}}} \gamma_\ell^{(t)} \cdot c_\ell$. There are three cases to consider:

**Case I:** If there exist $\alpha'_0, \ldots, \alpha'_{n_{\mathsf{pp}}} \in \mathbb{Z}_N$ such that $\left(\alpha'_0, \ldots, \alpha'_{n_{\mathsf{pp}}}, \beta_1^{(t)}, \ldots, \beta_{n_{\mathsf{sk}}}^{(t)}\right) \in \mathsf{span}(\mathcal{L}_{id})$, then $\alpha'_0 \cdot 1 + \sum_{\ell=1}^{n_{\mathsf{pp}}} \alpha'_\ell \cdot pp_\ell + \sum_{\ell=1}^{n_{\mathsf{sk}}} \beta_\ell^{(t)} \cdot sk_{id,\ell} = 0$. Therefore, $\alpha_0^{(t)} \cdot 1 + \sum_{\ell=1}^{n_{\mathsf{pp}}} \alpha_\ell^{(t)} \cdot pp_\ell + \sum_{\ell=1}^{n_{\mathsf{sk}}} \beta_\ell^{(t)} \cdot sk_{id,\ell} +$

$\sum_{\ell=1}^{n_{\mathsf{ct}}} \gamma_i^{(t)} \cdot c_i = 0$ if and only if $(\alpha_0^{(t)} - \alpha'_0) \cdot 1 + \sum_{\ell=1}^{n_{\mathsf{pp}}} (\alpha_\ell^{(t)} - \alpha'_\ell) \cdot pp_\ell + \sum_{\ell=1}^{n_{\mathsf{ct}}} \gamma_\ell^{(t)} \cdot c_\ell = 0$, and thus the emulation obtains the correct answer to the equality query $(i,j)$.

**Case II:** If the equality query $(i,j)$ is negatively answered in $\mathsf{Dec}$'s computation, and there do not exist $\alpha'_0, \ldots, \alpha'_{n_{\mathsf{pp}}} \in \mathbb{Z}_N$ such that $\left(\alpha'_0, \ldots, \alpha'_{n_{\mathsf{pp}}}, \beta_1^{(t)}, \ldots, \beta_{n_{\mathsf{sk}}}^{(t)}\right) \in \mathsf{span}(\mathcal{L}_{id})$, then it is also answered negatively in $\widetilde{\mathsf{Dec}}$'s computation.

**Case III:** If the equality query $(i,j)$ is positively answered in $\mathsf{Dec}$'s computation, and there do not exist $\alpha'_0, \ldots, \alpha'_{n_{\mathsf{pp}}} \in \mathbb{Z}_N$ such that $\left(\alpha'_0, \ldots, \alpha'_{n_{\mathsf{pp}}}, \beta_1^{(t)}, \ldots, \beta_{n_{\mathsf{sk}}}^{(t)}\right) \in \mathsf{span}(\mathcal{L}_{id})$, then the equality query $(i,j)$ is negatively answered in $\widetilde{\mathsf{Dec}}$'s computation. However, we show that this case occurs with probability at most $1/p(\lambda)$.

Recall that a ciphertext $c \leftarrow \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, id, b)$ is of the form $c = (c_1, \ldots, c_{n_{\mathsf{ct}}}, c_{\mathsf{str}})$, where $c_1, \ldots, c_{n_{\mathsf{ct}}}$ are group elements and $c_{\mathsf{str}}$ is an explicit string. Since the only group elements that are given as input to the computation $\mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, id, b)$ are $1, pp_1, \ldots, pp_{n_{\mathsf{pp}}}$, then each $c_v$ is of the form $c_v = \delta_{v,0} \cdot 1 + \sum_{\ell=1}^{n_{\mathsf{pp}}} \delta_{v,\ell} \cdot pp_\ell$, for coefficients $\delta_{v,0}, \ldots, \delta_{v,n_{\mathsf{pp}}} \in \mathbb{Z}_N$ that are determined by the group-operation queries issued during the computation $\mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, id, b)$. Therefore,

$$\begin{aligned} V_i - V_j &= \alpha_0^{(t)} \cdot 1 + \sum_{\ell=1}^{n_{\mathsf{pp}}} \alpha_\ell^{(t)} \cdot pp_\ell + \sum_{\ell=1}^{n_{\mathsf{sk}}} \beta_\ell^{(t)} \cdot sk_{id,\ell} + \sum_{\ell=1}^{n_{\mathsf{ct}}} \gamma_\ell^{(t)} \cdot c_\ell \\ &= \left( \alpha_0^{(t)} + \sum_{v=1}^{n_{\mathsf{ct}}} \cdot \delta_{v,0} \right) \cdot 1 + \sum_{\ell=1}^{n_{\mathsf{pp}}} \left( \alpha_\ell^{(t)} + \sum_{v=1}^{n_{\mathsf{ct}}} \cdot \delta_{v,\ell} \right) \cdot pp_\ell + \sum_{\ell=1}^{n_{\mathsf{sk}}} \beta_\ell^{(t)} \cdot sk_{id,\ell}. \end{aligned}$$

Now, in this case there do not exist $\alpha'_0, \ldots, \alpha'_{n_{\mathsf{pp}}} \in \mathbb{Z}_N$ such that $\left(\alpha'_0, \ldots, \alpha'_{n_{\mathsf{pp}}}, \beta_1^{(t)}, \ldots, \beta_{n_{\mathsf{sk}}}^{(t)}\right) \in \mathsf{span}(\mathcal{L}_{id})$, and therefore in particular $\left(\alpha'_0, \ldots, \alpha'_{n_{\mathsf{pp}}}, \beta_1^{(t)}, \ldots, \beta_{n_{\mathsf{sk}}}^{(t)}\right) \notin \mathsf{span}(\bigcup_{i \in [M-1]} \mathcal{L}_{id,b,i})$ for the specific choice of $\alpha'_\ell = \left(\alpha_\ell^{(t)} + \sum_{v=1}^{n_{\mathsf{ct}}} \cdot \delta_{v,\ell}\right)$ for every $\ell \in \{0, \ldots, n_{\mathsf{pp}}\}$. That is, this implies that for the computation $\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_{id}, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, id, b))$ it holds that

$$\mathcal{EQ}\left(\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_{id}, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, id, b))\right) \nsubseteq \mathsf{span}\left( \bigcup_{i \in [M-1]} \mathcal{L}_{id,b,i} \right).$$

Applying Lemma 3.2 with the linear subspace $V \subseteq \mathbb{Z}_N^{n_{\mathsf{pp}}+n_{\mathsf{sk}}+1}$ defined as

$$V = \left\{ \left(\alpha_0, \ldots, \alpha_{n_{\mathsf{pp}}}, \beta_1, \ldots, \beta_{n_{\mathsf{sk}}}\right) \in \mathbb{Z}_N^{n_{\mathsf{pp}}+n_{\mathsf{sk}}+1} \;\middle|\; \alpha_0 \cdot 1 + \sum_{\ell=1}^{n_{\mathsf{pp}}} \alpha_\ell \cdot pp_\ell + \sum_{\ell=1}^{n_{\mathsf{sk}}} \beta_\ell \cdot sk_{id,\ell} = 0 \right\},$$

which is of dimension at most $n_{\mathsf{pp}} + n_{\mathsf{sk}}$ since $(1, pp_1, \ldots, pp_{n_{\mathsf{pp}}}, sk_{id,1}, \ldots, sk_{id,n_{\mathsf{sk}}})$ is a non-zero vector, and with random variables $X_1, \ldots, X_M$ that are independently sampled from the distribution

$\mathcal{EQ}\left(\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_{id}, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, id, b))\right)$, we obtain from Eq. (3.1) that

$$\begin{aligned}
&\Pr\left[\widetilde{\mathsf{Dec}}^{\mathcal{O}}(\mathsf{pp}, \widetilde{\mathsf{sk}_{id}}, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, id, b)) \neq b\right] \\
&\qquad \leq \epsilon(\lambda) + \Pr\left[\mathcal{EQ}\left(\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_{id}, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, id, b))\right) \nsubseteq \mathsf{span}\left(\bigcup_{i \in [M-1]} \mathcal{L}_{id,b,i}\right)\right] \\
&\qquad = \epsilon(\lambda) + \Pr\left[X_M \nsubseteq \mathsf{span}\left(X_1 \cup \cdots \cup X_{M-1}\right)\right] \\
&\qquad \leq \epsilon(\lambda) + \frac{\mathsf{dim}V}{M(\lambda)} \\
&\qquad \leq \epsilon(\lambda) + \frac{n_{\mathsf{pp}}(\lambda) + n_{\mathsf{sk}}(\lambda)}{M(\lambda)} \\
&\qquad = \epsilon(\lambda) + \frac{1}{p(\lambda)}.
\end{aligned}$$

■

## 4 Attacking Generic-Group IBE Schemes

In this section we present a generic-group adversary that breaks the security of any generic-group identity-based encryption scheme whose secret keys do not contain group elements, and that supports more identities than the number of group elements included in its public parameters. We prove the following theorem:

**Theorem 4.1.** *Let $n_{\mathsf{pp}}(\lambda)$ be a function of the security parameter $\lambda \in \mathbb{N}$. Let $\mathcal{IBE}$ be a secure generic-group identity-based encryption scheme over an identity space $\mathcal{ID} = \{\mathcal{ID}_\lambda\}_{\lambda \in \mathbb{N}}$ with decryption error $\epsilon(\lambda) \leq 1/160(n_{\mathsf{pp}}(\lambda)+1)$, whose public parameters consist of $n_{\mathsf{pp}}(\lambda)$ group elements, and whose secret keys do not contain group elements. Then, $|\mathcal{ID}_\lambda| \leq n_{\mathsf{pp}}(\lambda)$ for all sufficiently large $\lambda \in \mathbb{N}$.*

Regarding the decryption error $\epsilon(\lambda) \leq 1/160(n_{\mathsf{pp}}(\lambda)+1)$ considered in the above theorem, recall that our transformation from Section 3 leads to an arbitrary polynomially-small increase in the scheme's decryption error.

**Preliminaries.** Recall that for any generic-group algorithm $A$ that receives as input group elements $g_1, \ldots, g_k$ and an explicit string $\mathsf{str}$ we let $\mathcal{EQ}\left(A^{\mathcal{O}}(\widehat{g_1}, \ldots, \widehat{g_k}, \mathsf{str})\right)$ denote the random variable corresponding to the set of all $(k+1)$-dimensional vectors over $\mathbb{Z}_N$ resulting from the positively-answered equality queries in the computation $A^{\mathcal{O}}(\widehat{g_1}, \ldots, \widehat{g_k}, \mathsf{str})$ (see Section 3 for the more formal definition). Our proof relies on the following lemma (which is proved in Appendix B):

**Lemma 4.2.** *Let $k \geq 1$, and let $X_1, \ldots, X_k$ be random variables over subsets of a linear vector space $V$ of dimension $\mathsf{dim}V$. Let $Y$ be distributed uniformly over $\{1, \ldots, k\}$ and independent of $X_1, \ldots, X_k$. Denote by $\mathsf{GoodSpan}$ the set of all $(i, U_1, \ldots, U_k) \subseteq [k] \times \left(2^V\right)^k$ for which*

$$\Pr_{X_1,\ldots,X_k,Y}\left[X_Y \subseteq \mathsf{span}\left(X_1 \cup \cdots \cup X_{Y-1}\right) \mid Y = i, X_1 = U_1, \ldots, X_{i-1} = U_{i-1}\right] \geq \frac{k - \mathsf{dim}V}{2k}.$$

*Then,*

$$\Pr_{X_1,\ldots,X_k,Y}\left[(Y, X_1, \ldots, X_k) \in \mathsf{GoodSpan}\right] \geq \frac{k - \mathsf{dim}V}{2k}.$$

The remainder of this section consists of the proof of Theorem 4.1.

**Proof of Theorem 4.1.** Let $\mathcal{IBE}$ be a generic-group identity-based encryption scheme over an identity space $\mathcal{ID} = \{\mathcal{ID}_\lambda\}_{\lambda\in\mathbb{N}}$ with decryption error $\epsilon(\lambda) \leq 1/160(n_{\mathsf{pp}}+1)$, whose public parameters consist of $n_{\mathsf{pp}} = n_{\mathsf{pp}}(\lambda)$ group elements, and whose secret keys do not contain group elements. Assume toward a contradiction that $|\mathcal{ID}_\lambda| \geq n_{\mathsf{pp}}+1$ for infinitely many values of $\lambda \in \mathbb{N}$, and assume without loss of generality that $\{1,\dots,n_{\mathsf{pp}}+1\} \subseteq \mathcal{ID}_\lambda$ for any such $\lambda \in \mathbb{N}$. We present a generic-group adversary $\mathcal{A}$ that issues a number of queries which is polynomial in $\lambda$, $n_{\mathsf{pp}}$, and in the number of queries issued by $\mathsf{Enc}$ and $\mathsf{Dec}$, and for which $\left|\Pr\left[\mathsf{Expt}_{\mathcal{IBE},\mathcal{A}}(\lambda) = 1\right] - 1/2\right|$ is non-negligible for any such $\lambda \in \mathbb{N}$ (recall Definition 2.3 describing the experiment $\mathsf{Expt}_{\mathcal{IBE},\mathcal{A}}$). The adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ is defined as follows:

**Our adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$**

**The algorithm $\mathcal{A}_1^{\mathcal{O}}(1^\lambda, N)$:**

1. Sample $i \leftarrow \{1,\dots,n_{\mathsf{pp}}+1\}$, and output the challenge identity $id^* = i$, the identities $(1,\dots,i-1)$ for which secret keys will be provided to $\mathcal{A}_2$, and the state $\mathsf{state} = (1^\lambda, N, i)$.

**The algorithm $\mathcal{A}_2^{\mathcal{O}}(\mathsf{state}, \mathsf{pp}, c^*, \mathsf{sk}_1, \dots, \mathsf{sk}_{i-1})$:**

1. Let $\mathsf{pp} = (pp_1,\dots,pp_{n_{\mathsf{pp}}}, \mathsf{pp}_{\mathsf{str}})$ for group elements $pp_1,\dots,pp_{n_{\mathsf{pp}}}$ and an explicit string $\mathsf{pp}_{\mathsf{str}}$ (and recall that $\mathsf{sk}_1,\dots,\mathsf{sk}_{i-1}$ are explicit strings).

**[Part I: Using $\mathsf{sk}_1,\dots,\mathsf{sk}_{i-1}$ for learning information on $\mathsf{pp}$]**

2. For each $j \in \{1,\dots,i-1\}$ perform the following steps:
   (a) Initialize a set $\mathcal{E}_j = \emptyset$ of $(n_{\mathsf{pp}}+1)$-dimensional vectors over $\mathbb{Z}_N$.
   (b) For each message $b \in \{0,1\}$ repeat the following step for $8(n_{\mathsf{pp}}+1)$ iterations:
   Compute $\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_j, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, j, b))$ using fresh randomness for $\mathsf{Enc}$ and $\mathsf{Dec}$, and update $\mathcal{E}_j \leftarrow \mathcal{E}_j \cup \mathcal{EQ}\left(\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, sk_j, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, j, b))\right)$.
   (c) Emulate a fresh oracle $\widehat{\mathcal{O}}$ in order to find $(\mathsf{pp}^*, \mathsf{msk}^*, \mathsf{sk}_j^*)$, where $\mathsf{pp}^* = (pp_1^*,\dots,pp_{n_{\mathsf{pp}}}^*, \mathsf{pp}_{\mathsf{str}}^*)$ for group elements $pp_1^*,\dots,pp_{n_{\mathsf{pp}}}^*$ and explicit strings $\mathsf{pp}_{\mathsf{str}}^*$, $\mathsf{msk}^*$ and $\mathsf{sk}_j^*$, subject to the following requirements:
      i. $(\mathsf{pp}^*, \mathsf{msk}^*)$ and $\mathsf{sk}_j^*$ are in the supports of $\mathsf{Setup}^{\widehat{\mathcal{O}}}(1^\lambda, N)$ and $\mathsf{KG}^{\widehat{\mathcal{O}}}(\mathsf{pp}^*, \mathsf{msk}^*, j)$, respectively.
      ii. $\mathsf{pp}_{\mathsf{str}}^* = \mathsf{pp}_{\mathsf{str}}$.
      iii. For every $(\alpha_0,\dots,\alpha_{n_{\mathsf{pp}}}) \in \mathcal{E}_1 \cup \dots \cup \mathcal{E}_{j-1}$ it holds that $\alpha_0 \cdot 1 + \sum_{\ell=1}^{n_{\mathsf{pp}}} \alpha_\ell \cdot pp_\ell^* = 0$ (i.e., $\mathsf{pp}^*$ satisfies the constraints induced by $\mathcal{E}_1 \cup \dots \cup \mathcal{E}_{j-1}$).
      iv. For each $b \in \{0,1\}$ it holds that $\Pr\left[\mathsf{Dec}^{\widehat{\mathcal{O}}}(\mathsf{pp}^*, \mathsf{sk}_j^*, \mathsf{Enc}^{\widehat{\mathcal{O}}}(\mathsf{pp}^*, j, b)) = b\right] \geq 19/20$, where the probability is taken over the internal randomness of $\mathsf{Enc}$ and $\mathsf{Dec}$ (i.e., the decryption error of $\mathsf{sk}_j^*$ is at most 1/20).
      v. For each $b \in \{0,1\}$ it holds that
      $$\Pr\left[\mathcal{EQ}\left(\mathsf{Dec}^{\widehat{\mathcal{O}}}(\mathsf{pp}^*, sk_j^*, \mathsf{Enc}^{\widehat{\mathcal{O}}}(\mathsf{pp}^*, j, b))\right) \nsubseteq \mathsf{span}\left(\mathcal{E}_1 \cup \dots \cup \mathcal{E}_{j-1}\right)\right] \leq \frac{1}{5},$$
      where the probability is taken over the internal randomness of $\mathsf{Enc}$ and $\mathsf{Dec}$.
   (d) If such $(\mathsf{msk}^*, \mathsf{pp}^*, sk_j^*)$ are found then for each message $b \in \{0,1\}$ repeat the following step for $8(n_{\mathsf{pp}}+1)$ iterations:
   Compute $\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_j^*, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, j, b))$ using fresh randomness for $\mathsf{Enc}$ and $\mathsf{Dec}$, and update $\mathcal{E}_j \leftarrow \mathcal{E}_j \cup \mathcal{EQ}\left(\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, sk_j^*, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, j, b))\right)$.

**[Part II: Constructing an alternative $\mathsf{sk}_i^*$ for decrypting the challenge ciphertext]**

3. Emulate a fresh oracle $\widehat{\mathcal{O}}$ in order to find $(\mathsf{pp}^*, \mathsf{msk}^*, \mathsf{sk}_i^*)$, where $\mathsf{pp}^* = (pp_1^*, \ldots, pp_{n_{\mathsf{pp}}}^*, \mathsf{pp}_{\mathsf{str}}^*)$ for group elements $pp_1^*, \ldots, pp_{n_{\mathsf{pp}}}^*$ and explicit strings $\mathsf{pp}_{\mathsf{str}}^*$, $\mathsf{msk}^*$ and $\mathsf{sk}_i^*$, subject to the following requirements:

   (a) $(\mathsf{pp}^*, \mathsf{msk}^*)$ and $\mathsf{sk}_i^*$ are in the supports of $\mathsf{Setup}^{\widehat{\mathcal{O}}}(1^\lambda, N)$ and $\mathsf{KG}^{\widehat{\mathcal{O}}}(\mathsf{pp}^*, \mathsf{msk}^*, i)$, respectively.

   (b) $\mathsf{pp}_{\mathsf{str}}^* = \mathsf{pp}_{\mathsf{str}}$.

   (c) For every $(\alpha_0, \ldots, \alpha_{n_{\mathsf{pp}}}) \in \mathcal{E}_1 \cup \cdots \cup \mathcal{E}_{i-1}$ it holds that $\alpha_0 \cdot 1 + \sum_{\ell=1}^{n_{\mathsf{pp}}} \alpha_\ell \cdot pp_\ell^* = 0$ (i.e., $\mathsf{pp}^*$ satisfies the constraints induced by $\mathcal{E}_1 \cup \cdots \cup \mathcal{E}_{i-1}$).

   (d) For each $b \in \{0,1\}$ it holds that $\Pr\left[\mathsf{Dec}^{\widehat{\mathcal{O}}}(\mathsf{pp}^*, \mathsf{sk}_i^*, \mathsf{Enc}^{\widehat{\mathcal{O}}}(\mathsf{pp}^*, i, b)) = b\right] \geq 19/20$ , where the probability is taken over the internal randomness of $\mathsf{Enc}$ and $\mathsf{Dec}$ (i.e., the decryption error of $\mathsf{sk}_i^*$ is at most $1/20$).

   (e) For each $b \in \{0,1\}$ it holds that

   $$\Pr\left[\mathcal{EQ}\left(\mathsf{Dec}^{\widehat{\mathcal{O}}}(\mathsf{pp}^*, sk_i^*, \mathsf{Enc}^{\widehat{\mathcal{O}}}(\mathsf{pp}^*, i, b))\right) \nsubseteq \mathsf{span}\left(\mathcal{E}_1 \cup \cdots \cup \mathcal{E}_{i-1}\right)\right] \leq \frac{1}{5},$$

   where the probability is taken over the internal randomness of $\mathsf{Enc}$ and $\mathsf{Dec}$.

4. If such $(\mathsf{msk}^*, \mathsf{pp}^*, sk_i^*)$ are not found then sample and output $b' \leftarrow \{0,1\}$.

5. If such $(\mathsf{msk}^*, \mathsf{pp}^*, sk_i^*)$ are found then compute $\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, sk_i^*, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, i, b))$ for $\lambda$ times, where each computation uses fresh randomness for $\mathsf{Enc}$, $\mathsf{Dec}$ and $b$, and count the number of times in which decryption was correct. If decryption was correct less than $\lambda \cdot \frac{11}{20} \cdot \left(1 - \frac{1}{20}\right)$ times, then sample and output $b' \leftarrow \{0,1\}$.

6. Compute and output $b' \leftarrow \mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, sk_i^*, c^*)$.

In what follows we first analyze the number of queries issued by $\mathcal{A}$, and then analyze its success probability. It terms of oracle queries (i.e., group-operations queries and equality queries), note that $\mathcal{A}_1$ does not issue any queries, and that $\mathcal{A}_2$ issues queries only in Steps 2(b), 2(d), 5 and 6. These queries result from invoking the algorithms $\mathsf{Enc}$ and $\mathsf{Dec}$, where Steps 2(b) and 2(d) consist of at most $O((n_{\mathsf{pp}})^2)$ such invocations, Step 5 consists of $\lambda$ such invocations, and Step 6 consists of one such invocation.

For analyzing $\mathcal{A}$'s success probability, fix a security parameter $\lambda \in \mathbb{N}$, a prime integer $N$ that is produced by $\mathsf{PrimeGen}(1^\lambda)$, and a pair $(\mathsf{msk}, \mathsf{pp})$ that is produced by $\mathsf{Setup}^{\mathcal{O}}(1^\lambda)$. Our proof relies on the following notation:

- The experiment $\mathsf{Expt}_{\mathcal{IBE},\mathcal{A}}$ and the description of our adversary define the random variables $\mathsf{sk}_1, \ldots, \mathsf{sk}_{i-1}$ corresponding to the secret keys that $\mathcal{A}_2$ is given as input. For our analysis, we additionally consider the random variables $\mathsf{sk}_i, \ldots, \mathsf{sk}_{n_{\mathsf{pp}}+1}$ that are independently sampled by computing $\mathsf{sk}_j \leftarrow \mathsf{KG}^{\mathcal{O}}(\mathsf{pp}, \mathsf{msk}, j)$ for each $j \in \{i, \ldots, n_{\mathsf{pp}} + 1\}$.

- We denote by $Y$ the random variable corresponding to the choice of the challenge identity $i \leftarrow \{1, \ldots, n_{\mathsf{pp}} + 1\}$ by $\mathcal{A}_1$.

- For each $j \in \{1, \ldots, n_{\mathsf{pp}} + 1\}$ we let $\mathcal{E}_j = \cup_{v=1}^{8(n_{\mathsf{pp}}+1)} \mathcal{E}_{j,v}$, where each $\mathcal{E}_{j,v}$ denotes the random variable corresponding to the set of vectors of coefficients of the equations found in one iteration of Step 2(b) and of Step 2(d). More specifically, each $\mathcal{E}_{j,v}$ is sampled from the distribution

$$\mathcal{EQ}\left(\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp},\mathsf{sk}_j,\mathsf{Enc}^{\mathcal{O}}(\mathsf{pp},j,0))\right)\cup\mathcal{EQ}\left(\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp},\mathsf{sk}_j,\mathsf{Enc}^{\mathcal{O}}(\mathsf{pp},j,1))\right) \\ \cup\,\mathcal{EQ}\left(\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp},sk^*_j,\mathsf{Enc}^{\mathcal{O}}(\mathsf{pp},j,0))\right)\cup\mathcal{EQ}\left(\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp},sk^*_j,\mathsf{Enc}^{\mathcal{O}}(\mathsf{pp},j,0))\right),$$

where if $\mathcal{A}_2$ does not find a suitable $sk^*_j$ in Step 2(c), then we define

$$\mathcal{EQ}\left(\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp},sk^*_j,\mathsf{Enc}^{\mathcal{O}}(\mathsf{pp},i,0))\right)=\mathcal{EQ}\left(\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp},sk^*_j,\mathsf{Enc}^{\mathcal{O}}(\mathsf{pp},i,0))\right)=\emptyset.$$

- We denote by $\mathsf{GoodSpan}$ the set of all $(i,U_1,\dots,U_{n_{\mathsf{pp}}+1})\in\{1,\dots,n_{\mathsf{pp}}+1\}\times\left(2^{\mathbb{Z}_N^{n_{\mathsf{pp}}+1}}\right)^{n_{\mathsf{pp}}+1}$ for which

$$\Pr[\mathcal{E}_Y\subseteq\mathsf{span}\left(\mathcal{E}_1\cup\cdots\cup\mathcal{E}_{Y-1}\right)\mid Y=i,\mathcal{E}_1=U_1,\dots\mathcal{E}_{i-1}=U_{i-1}]\geq\frac{1}{2(n_{\mathsf{pp}}+1)}.$$

  For avoiding additional notation, we abuse notation and denote by $\mathsf{GoodSpan}$ the event in which $(Y,\mathcal{E}_1,\dots,\mathcal{E}_{n_{\mathsf{pp}}+1})\in\mathsf{GoodSpan}$.

**Claim 4.3.** $\Pr[\mathsf{GoodSpan}]\geq\frac{1}{2(n_{\mathsf{pp}}+1)}$.

**Proof.** This is direct application of Lemma 4.2 with $k=n_{\mathsf{pp}}+1$, $(X_1,\dots,X_k)=(\mathcal{E}_1,\dots,\mathcal{E}_k)$, $Y$ as defined above, and

$$V=\left\{(\alpha_0,\dots,\alpha_{n_{\mathsf{pp}}})\in\mathbb{Z}_N^{n_{\mathsf{pp}}+1}\;\middle|\;\alpha_0\cdot 1+\sum_{\ell=1}^{n_{\mathsf{pp}}}\alpha_\ell\cdot pp_\ell=0\right\}.$$

Note that since $(1,pp_1,\dots,pp_{n_{\mathsf{pp}}})$ is a non-zero vector then $\mathsf{dim}V\leq n_{\mathsf{pp}}$. ■

For the next claim, we denote by $\mathsf{FindKey}$ the event that $\mathcal{A}_2$ finds $(\mathsf{msk}^*,\mathsf{pp}^*,\mathsf{sk}^*_i)$ in Step 3 that satisfies the required properties.

**Claim 4.4.** $\mathsf{GoodSpan}\subseteq\mathsf{FindKey}$.

**Proof.** We show that whenever the event $\mathsf{GoodSpan}$ occurs, then $\mathsf{msk}$, $\mathsf{pp}$, and at least one $\mathsf{sk}_Y$ in the support of $\mathsf{KG}^{\mathcal{O}}(\mathsf{pp},\mathsf{msk},Y)$ already satisfy the the required properties. Therefore, in particular, $\mathcal{A}_2$ finds some $(\mathsf{msk}^*,\mathsf{pp}^*,\mathsf{sk}^*_i)$ in Step 3 that satisfies the required properties. Properties (a), (b) and (c) are trivially satisfied by $(\mathsf{msk},\mathsf{pp},\mathsf{sk}_Y)$ for any $\mathsf{sk}_Y$ in the support of $\mathsf{KG}^{\mathcal{O}}(\mathsf{pp},\mathsf{msk},Y)$. In what follows we show that properties (d) and (e) are satisfied by at least one $\mathsf{sk}_Y$ in the support of $\mathsf{KG}^{\mathcal{O}}(\mathsf{pp},\mathsf{msk},Y)$.

The decryption error of the scheme is at most $\frac{1}{160(n_{\mathsf{pp}}+1)}$. Therefore, for any value of $Y$ it holds that

$$\Pr_{\mathsf{KG},\mathsf{Enc},\mathsf{Dec}}\left[\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp},\mathsf{sk}_Y,\mathsf{Enc}^{\mathcal{O}}(\mathsf{pp},Y,b))=b\right]\geq 1-\frac{1}{160(n_{\mathsf{pp}}+1)}$$

where $\mathsf{sk}_Y\leftarrow\mathsf{KG}^{\mathcal{O}}(\mathsf{pp},\mathsf{msk},Y)$, and the probability is taken over the internal randomness of the algorithms $\mathsf{KG}$, $\mathsf{Enc}$ and $\mathsf{Dec}$. The above holds for any value of $Y$ and independently of $\mathcal{E}_1\cup\cdots\cup\mathcal{E}_{Y-1}$, and therefore

$$\Pr_{\mathsf{KG},\mathsf{Enc},\mathsf{Dec}}\left[\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp},\mathsf{sk}_Y,\mathsf{Enc}^{\mathcal{O}}(\mathsf{pp},Y,b))=b\mid\mathsf{GoodSpan}\right]\geq 1-\frac{1}{160(n_{\mathsf{pp}}+1)},$$

where $\mathsf{sk}_Y \leftarrow \mathsf{KG}^{\mathcal{O}}(\mathsf{pp}, \mathsf{msk}, Y)$, and the probability is taken over the internal randomness of the algorithms $\mathsf{KG}$, $\mathsf{Enc}$ and $\mathsf{Dec}$. Denote by $\mathsf{SkSmallError}_Y$ the set of all outputs $\mathsf{sk}_Y$ of $\mathsf{KG}^{\mathcal{O}}(\mathsf{pp}, \mathsf{msk}, Y)$ for which

$$\Pr_{\mathsf{Enc},\mathsf{Dec}}\left[\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_Y, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, Y, b)) = b \mid \mathsf{GoodSpan}\right] \geq \frac{19}{20},$$

where now the probability is taken only over the internal randomness of the algorithms $\mathsf{Enc}$ and $\mathsf{Dec}$. Then,

$$\begin{aligned}1 - \frac{1}{160(n_{\mathsf{pp}}+1)} &\leq \Pr\left[\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_Y, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, Y, b)) = b \mid \mathsf{GoodSpan}\right] \\ &\leq \Pr[\mathsf{sk}_Y \in \mathsf{SkSmallError}_Y \mid \mathsf{GoodSpan}] \cdot 1 \\ &\quad + (1 - \Pr[\mathsf{sk}_Y \in \mathsf{SkSmallError}_Y \mid \mathsf{GoodSpan}]) \cdot \frac{19}{20}.\end{aligned}$$

Therefore,

$$\Pr[\mathsf{sk}_Y \in \mathsf{SkSmallError}_Y \mid \mathsf{GoodSpan}] \geq 1 - \frac{1}{8(n_{\mathsf{pp}}+1)}. \tag{4.1}$$

Similarly, denote by $\mathsf{SkGood}_Y$ the set of all outputs $\mathsf{sk}_Y$ of $\mathsf{KG}^{\mathcal{O}}(\mathsf{pp}, \mathsf{msk}, Y)$ for which

$$\Pr[\mathcal{E}_Y \subseteq \mathsf{span}\,(\mathcal{E}_1 \cup \cdots \cup \mathcal{E}_{Y-1}) \mid (Y, \mathcal{E}_1, \ldots, \mathcal{E}_{n_{\mathsf{pp}}+1}) \in \mathsf{GoodSpan},\, \mathsf{sk}_Y] \geq \frac{1}{4(n_{\mathsf{pp}}+1)}.$$

Then, from the definition of $\mathsf{GoodSpan}$ we obtain

$$\begin{aligned}\frac{1}{2(n_{\mathsf{pp}}+1)} &\leq \Pr[\mathcal{E}_Y \subseteq \mathsf{span}\,(\mathcal{E}_1 \cup \cdots \cup \mathcal{E}_{Y-1}) \mid (Y, \mathcal{E}_1, \ldots, \mathcal{E}_{n_{\mathsf{pp}}+1}) \in \mathsf{GoodSpan}] \\ &\leq \Pr[\mathsf{SkGood}_Y \mid \mathsf{GoodSpan}] \cdot 1 + (1 - \Pr[\mathsf{SkGood}_Y \mid \mathsf{GoodSpan}]) \cdot \frac{1}{4(n_{\mathsf{pp}}+1)}\end{aligned}$$

and therefore

$$\Pr[\mathsf{sk}_Y \in \mathsf{SkGood}_Y \mid (Y, \mathcal{E}_1, \ldots, \mathcal{E}_{n_{\mathsf{pp}}+1}) \in \mathsf{GoodSpan}] \geq \frac{1}{4(n_{\mathsf{pp}}+1)}. \tag{4.2}$$

Therefore, combining Eq. (4.1) and (4.2) we obtain

$$\begin{aligned}&\Pr[\mathsf{sk}_Y \in \mathsf{SkGood}_Y \cap \mathsf{SkSmallError}_Y \mid (Y, \mathcal{E}_1, \ldots, \mathcal{E}_{n_{\mathsf{pp}}+1}) \in \mathsf{GoodSpan}] \\ &\qquad \geq \frac{1}{4(n_{\mathsf{pp}}+1)} - \frac{1}{8(n_{\mathsf{pp}}+1)} \\ &\qquad = \frac{1}{8(n_{\mathsf{pp}}+1)}. \end{aligned} \tag{4.3}$$

Note that property (d) is satisfied by any $\mathsf{sk}_Y \in \mathsf{SkSmallError}_Y$. We will now show that property (e) is satisfied by any $\mathsf{sk}_Y \in \mathsf{SkGood}_Y$ conditioned on $\mathsf{GoodSpan}$, which together with Eq. (4.3) (and the fact that $\Pr[\mathsf{GoodSpan}] > 0$ as shown in Claim 4.3) settles the proof.

Recall that $\mathcal{E}_Y = \cup_{v=1}^{8(n_{\mathsf{pp}}+1)} \mathcal{E}_{Y,v}$ where $\{\mathcal{E}_{Y,v}\}_{v=1}^{8(n_{\mathsf{pp}}+1)}$ are identically distributed and independent given $\mathcal{E}_1, \ldots, \mathcal{E}_{Y-1}$ and $\mathsf{sk}_Y$. Therefore, the definition of $\mathsf{SkGood}_Y$ implies that

$$\begin{aligned}
\frac{1}{4(n_{\mathsf{pp}}+1)} &\leq \Pr[\mathcal{E}_Y \subseteq \mathsf{span}\left(\mathcal{E}_1 \cup \cdots \cup \mathcal{E}_{Y-1}\right) \mid (Y, \mathcal{E}_1, \ldots, \mathcal{E}_{n_{\mathsf{pp}}+1}) \in \mathsf{GoodSpan},\ \mathsf{sk}_Y \in \mathsf{SkGood}_Y] \\
&= \Pr[\wedge_{v=1}^{8(n_{\mathsf{pp}}+1)} \left(\mathcal{E}_{Y,v} \subseteq \mathsf{span}\left(\mathcal{E}_1 \cup \cdots \cup \mathcal{E}_{Y-1}\right)\right) \mid (Y, \mathcal{E}_1, \ldots, \mathcal{E}_{n_{\mathsf{pp}}+1}) \in \mathsf{GoodSpan},\ \mathsf{sk}_Y \in \mathsf{SkGood}_Y] \\
&= \prod_{v=1}^{8(n_{\mathsf{pp}}+1)} \Pr[(\mathcal{E}_{Y,v} \subseteq \mathsf{span}\left(\mathcal{E}_1 \cup \cdots \cup \mathcal{E}_{Y-1}\right)) \mid (Y, \mathcal{E}_1, \ldots, \mathcal{E}_{n_{\mathsf{pp}}+1}) \in \mathsf{GoodSpan},\ \mathsf{sk}_Y \in \mathsf{SkGood}_Y] \\
&= \left(\Pr[(\mathcal{E}_{Y,1} \subseteq \mathsf{span}\left(\mathcal{E}_1 \cup \cdots \cup \mathcal{E}_{Y-1}\right)) \mid (Y, \mathcal{E}_1, \ldots, \mathcal{E}_{n_{\mathsf{pp}}+1}) \in \mathsf{GoodSpan},\ \mathsf{sk}_Y \in \mathsf{SkGood}_Y]\right)^{8(n_{\mathsf{pp}}+1)}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\Pr[(\mathcal{E}_{Y,1} \subseteq \mathsf{span}\left(\mathcal{E}_1 \cup \cdots \cup \mathcal{E}_{Y-1}\right)) \mid (Y, \mathcal{E}_1, \ldots, \mathcal{E}_k) \in \mathsf{GoodSpan},\ \mathsf{sk}_Y \in \mathsf{SkGood}_Y] &\geq \left(\frac{1}{4(n_{\mathsf{pp}}+1)}\right)^{\frac{1}{8(n_{\mathsf{pp}}+1)}} \\
&\geq \frac{4}{5}.
\end{aligned}$$

In addition, recall that,

$$\begin{aligned}
\mathcal{E}_{Y,v} = \mathcal{EQ}\left(\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_Y, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, Y, 0))\right) \cup \mathcal{EQ}\left(\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_Y, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, Y, 1))\right) \\
\cup\, \mathcal{EQ}\left(\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}^*_Y, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, Y, 0))\right) \cup \mathcal{EQ}\left(\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, sk^*_Y, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, Y, 0))\right),
\end{aligned}$$

Now, since for each $b \in \{0,1\}$, $\mathcal{EQ}\left(\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_Y, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, Y, b))\right) \subseteq \mathcal{E}_{Y,1}$, then for each $b \in \{0,1\}$ it holds that

$$\begin{aligned}
\Pr[\mathcal{EQ}\left(\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_Y, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, Y, b))\right) \subseteq \mathsf{span}\left(\mathcal{E}_1 \cup \cdots \cup \mathcal{E}_{i-1}\right) \mid \\
(Y, \mathcal{E}_1, \ldots, \mathcal{E}_{n_{\mathsf{pp}}+1}) \in \mathsf{GoodSpan},\ \mathsf{sk}_Y \in \mathsf{SkGood}_Y] \geq \frac{4}{5},
\end{aligned}$$

as required. ■

For the next claim, note that if the event GoodSpan occurs, then by Claim 4.4 the event FindKey occurs as well, and therefore $\mathsf{pp}^*$, $\mathsf{msk}^*$ and $\mathsf{sk}^*_i$ are well defined.

**Claim 4.5.** *For each $b \in \{0,1\}$ it holds that*

$$\Pr\left[\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}^*_i, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, i, b; r); r') = \mathsf{Dec}^{\widehat{\mathcal{O}}}(\mathsf{pp}^*, \mathsf{sk}^*_i, \mathsf{Enc}^{\widehat{\mathcal{O}}}(\mathsf{pp}^*, i, b; r); r') \,\middle|\, \mathsf{GoodSpan}\right] \geq \frac{3}{5},$$

*where the probability is taken over the internal randomness $r \in \{0,1\}^*$ and $r' \in \{0,1\}^*$ of* Enc *and* Dec*, respectively.*

**Proof.** Fix $b \in \{0,1\}$. The definition of the set GoodSpan, together with the fact that $\mathcal{E}_Y = \cup_{v=1}^{8(n_{\mathsf{pp}}+1)} \mathcal{E}_{Y,v}$ where $\{\mathcal{E}_{Y,v}\}_{v=1}^{8(n_{\mathsf{pp}}+1)}$ are identically distributed and independent given $\mathcal{E}_1, \ldots, \mathcal{E}_{Y-1}$ and $\mathsf{sk}_Y$, imply that

$$\begin{aligned}
\frac{1}{2(n_{\mathsf{pp}}+1)} &\leq \Pr[\mathcal{E}_Y \subseteq \mathsf{span}\left(\mathcal{E}_1 \cup \cdots \cup \mathcal{E}_{Y-1}\right) \mid (Y, \mathcal{E}_1, \ldots, \mathcal{E}_k) \in \mathsf{GoodSpan},\ \mathsf{sk}_Y] \\
&= \Pr[\wedge_{v=1}^{8(n_{\mathsf{pp}}+1)} \left(\mathcal{E}_{Y,j} \subseteq \mathsf{span}\left(\mathcal{E}_1 \cup \cdots \cup \mathcal{E}_{Y-1}\right)\right) \mid (Y, \mathcal{E}_1, \ldots, \mathcal{E}_k) \in \mathsf{GoodSpan},\ \mathsf{sk}_Y] \\
&= \prod_{v=1}^{8(n_{\mathsf{pp}}+1)} \Pr[(\mathcal{E}_{Y,j} \subseteq \mathsf{span}\left(\mathcal{E}_1 \cup \cdots \cup \mathcal{E}_{Y-1}\right)) \mid (Y, \mathcal{E}_1, \ldots, \mathcal{E}_k) \in \mathsf{GoodSpan},\ \mathsf{sk}_Y] \\
&= (\Pr[(\mathcal{E}_{Y,1} \subseteq \mathsf{span}\left(\mathcal{E}_1 \cup \cdots \cup \mathcal{E}_{Y-1}\right)) \mid (Y, \mathcal{E}_1, \ldots, \mathcal{E}_k) \in \mathsf{GoodSpan},\ \mathsf{sk}_Y])^{8(n_{\mathsf{pp}}+1)}.
\end{aligned}$$

Therefore,

$$\begin{aligned}\Pr[(\mathcal{E}_{Y,1} \subseteq \mathsf{span}\left(\mathcal{E}_1 \cup \dots \cup \mathcal{E}_{Y-1}\right)) \mid (Y, \mathcal{E}_1, \dots, \mathcal{E}_k) \in \mathsf{GoodSpan}, \mathsf{sk}_Y] &\geq \left(\frac{1}{2(n_{\mathsf{pp}}+1)}\right)^{\frac{1}{8(n_{\mathsf{pp}}+1)}} \\ &\geq \frac{4}{5}.\end{aligned}$$

Since

$$\begin{aligned}\mathcal{E}_{i,1} = \mathcal{EQ}\left(\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_i, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, i, 0))\right) \cup \mathcal{EQ}\left(\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_i, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, i, 1))\right) \\ \cup \mathcal{EQ}\left(\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}^*_i, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, i, 0))\right) \cup \mathcal{EQ}\left(\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}^*_i, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, i, 0))\right),\end{aligned}$$

then, in particular, it holds that

$$\Pr[\mathcal{EQ}\left(\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}^*_i, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, i, b))\right) \subseteq \mathsf{span}\left(\mathcal{E}_1 \cup \dots \cup \mathcal{E}_{i-1}\right) \mid \mathsf{GoodSpan}, \mathsf{sk}_i] \geq \frac{4}{5}.$$

Since $\mathsf{sk}^*_i$ and the randomness of $\mathsf{Enc}$ and $\mathsf{Dec}$ are independent of $\mathsf{sk}_i$, then also

$$\Pr[\mathcal{EQ}\left(\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}^*_i, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, i, b))\right) \subseteq \mathsf{span}\left(\mathcal{E}_1 \cup \dots \cup \mathcal{E}_{i-1}\right) \mid \mathsf{GoodSpan}] \geq \frac{4}{5}.$$

One of the requirements of $(\mathsf{msk}^*, \mathsf{pp}^*, \mathsf{sk}^*_i)$ is that

$$\Pr\left[\mathcal{EQ}\left(\mathsf{Dec}^{\widehat{\mathcal{O}}}(\mathsf{pp}^*, sk^*_i, \mathsf{Enc}^{\widehat{\mathcal{O}}}(\mathsf{pp}^*, i, b))\right) \subseteq \mathsf{span}\left(\mathcal{E}_1 \cup \dots \cup \mathcal{E}_{i-1}\right)\right] \geq \frac{4}{5},$$

where the probability is taken over the internal randomness of $\mathsf{Enc}$ and $\mathsf{Dec}$, and therefore

$$\begin{aligned}\Pr[\mathcal{EQ}\left(\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}^*_i, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, i, b; r); r')\right) \cup \mathcal{EQ}\left(\mathsf{Dec}^{\widehat{\mathcal{O}}}(\mathsf{pp}^*, sk^*_i, \mathsf{Enc}^{\widehat{\mathcal{O}}}(\mathsf{pp}^*, i, b; r); r')\right) \\ \subseteq \mathsf{span}\left(\mathcal{E}_1 \cup \dots \cup \mathcal{E}_{i-1}\right) \mid \mathsf{GoodSpan}] \geq \frac{4}{5} + \frac{4}{5} - 1 = \frac{3}{5},\end{aligned}$$

where the probability is taken over the internal randomness $r \in \{0,1\}^*$ and $r' \in \{0,1\}^*$ of $\mathsf{Enc}$ and $\mathsf{Dec}$, respectively. Now, for each such $r$ and $r'$ that satisfy

$$\mathcal{EQ}\left(\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}^*_i, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, i, b; r); r')\right) \cup \mathcal{EQ}\left(\mathsf{Dec}^{\widehat{\mathcal{O}}}(\mathsf{pp}^*, sk^*_i, \mathsf{Enc}^{\widehat{\mathcal{O}}}(\mathsf{pp}^*, i, b; r); r')\right) \subseteq \mathsf{span}\left(\mathcal{E}_1 \cup \dots \cup \mathcal{E}_{i-1}\right)$$

we claim that

$$\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}^*_i, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, i, b; r); r') = \mathsf{Dec}^{\widehat{\mathcal{O}}}(\mathsf{pp}^*, \mathsf{sk}^*_i, \mathsf{Enc}^{\widehat{\mathcal{O}}}(\mathsf{pp}^*, i, b; r); r').$$

The two computations have the same explicit inputs since $pp_{\mathsf{str}} = pp^*_{\mathsf{str}}$ and $\mathsf{sk}^*_i$ does not contain group elements. Assuming that the responses to the equality queries are consistent among the two computations up to a certain point, then both computations issue the exact same next equality query $(i_1, i_2)$. Let $V_{i_1}$ and $V_{i_2}$ denote the group elements that are located in the corresponding entries of the table $\mathbf{B}$ associated with oracle $\mathcal{O}$. Let $V^*_{i_1}$ and $V^*_{i_2}$ denote the group elements that are located in the corresponding entries of the table $\widehat{\mathbf{B}}$ associated with oracle $\widehat{\mathcal{O}}$. Let $V_{i_1} - V_{i_2} = \alpha_0 \cdot 1 + \sum_{r=1}^{n_{\mathsf{pp}}} \alpha_r \cdot pp_r$ for $\alpha_0, \dots, \alpha_r \in \mathbb{Z}_N$. Since the two computations are the same up to this point, $V^*_{i_1} - V^*_{i_2} = \alpha_0 \cdot 1 + \sum_{r=1}^{n_{\mathsf{pp}}} \alpha_r \cdot pp^*_r$.

On the one hand, if the equality query $(i_1, i_2)$ is answered positively in the computation $\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_i^*, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, i, b; r); r')$, then $(\alpha_0, \ldots, \alpha_{n_{\mathsf{pp}}}) \in \mathcal{EQ}\left(\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_i^*, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, i, b; r); r')\right)$ and therefore $(\alpha_0, \ldots, \alpha_{n_{\mathsf{pp}}}) \in \mathsf{span}\,(\mathcal{E}_1 \cup \cdots \cup \mathcal{E}_{i-1})$. But $\mathsf{pp}^*$ is chosen to satisfy $(\mathcal{E}_1 \cup \cdots \cup \mathcal{E}_{i-1})$, and so $\alpha_0 \cdot 1 + \sum_{r=1}^{n_{\mathsf{pp}}} \alpha_r \cdot pp_r^* = 0$, and this equality query is also answered positively by the computation $\mathsf{Dec}^{\widehat{\mathcal{O}}}(\mathsf{pp}^*, \mathsf{sk}_i^*, \mathsf{Enc}^{\widehat{\mathcal{O}}}(\mathsf{pp}^*, i, b; r); r')$.

On the other hand, if the equality query $(i_1, i_2)$ is answered positively by the computation $\mathsf{Dec}^{\widehat{\mathcal{O}}}(\mathsf{pp}^*, \mathsf{sk}_i^*, \mathsf{Enc}^{\widehat{\mathcal{O}}}(\mathsf{pp}^*, i, b; r); r')$, then an symmetric argument shows that this equality query is also answered positively by the computation $\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_i^*, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, i, b; r); r')$. ■

**Claim 4.6.** *For each $b \in \{0, 1\}$ it holds that*

$$\Pr\left[\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_i^*, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, i, b)) = b \;\middle|\; \mathsf{GoodSpan}\right] \geq \frac{11}{20}.$$

**Proof.** The event $\mathsf{GoodSpan}$ implies the event $\mathsf{FindKey}$, and therefore the secret key $\mathsf{sk}_i^*$ chosen in Step 3 satisfies

$$\Pr\left[\mathsf{Dec}^{\widehat{\mathcal{O}}}(\mathsf{pp}^*, \mathsf{sk}_i^*, \mathsf{Enc}^{\widehat{\mathcal{O}}}(\mathsf{pp}^*, i, b)) = b\right] \geq \frac{19}{20},$$

where the probability is taken over the internal randomness of the algorithms $\mathsf{Enc}$ and $\mathsf{Dec}$. Combining this with Claim 4.5 we obtain

$$\begin{aligned}
&\Pr\left[\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_i^*, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, i, b)) \neq b \;\middle|\; \mathsf{GoodSpan}\right] \\
&\quad \leq \Pr\left[\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_i^*, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, i, b; r); r') \neq \mathsf{Dec}^{\widehat{\mathcal{O}}}(\mathsf{pp}^*, \mathsf{sk}_i^*, \mathsf{Enc}^{\widehat{\mathcal{O}}}(\mathsf{pp}^*, i, b; r); r') \middle| \mathsf{GoodSpan}\right] \\
&\qquad + \Pr\left[\mathsf{Dec}^{\widehat{\mathcal{O}}}(\mathsf{pp}^*, \mathsf{sk}_i^*, \mathsf{Enc}^{\widehat{\mathcal{O}}}(\mathsf{pp}^*, i, b)) \neq b\right] \\
&\quad \leq \frac{2}{5} + \frac{1}{20} \\
&\quad = \frac{9}{20}.
\end{aligned}$$

■

For the following claims, we denote by $\mathsf{Pass}$ the event that $\mathsf{sk}_i^*$ passes the test in Step 5, and denote by $\mathsf{Win}$ the event in which $\mathsf{Expt}_{\mathcal{IBE},\mathcal{A}}(\lambda) = 1$.

**Claim 4.7.** $\Pr[\mathsf{Win} \mid \mathsf{GoodSpan}] \geq \frac{11}{20} \cdot \left(1 - e^{-\Omega(\lambda)}\right)$.

**Proof.** Claim 4.6 and Chernoff's bound imply that

$$\begin{aligned}
\Pr[\mathsf{Win} \mid \mathsf{GoodSpan}] &\geq \Pr[\mathsf{Win} \mid \mathsf{Pass} \cap \mathsf{GoodSpan}] \cdot \Pr[\mathsf{Pass} \mid \mathsf{GoodSpan}] \\
&\geq \frac{11}{20} \cdot \left(1 - e^{-\frac{(0.05)^2}{2} \cdot \frac{11}{20} \cdot \lambda}\right).
\end{aligned}$$

■

**Claim 4.8.** $\Pr[\mathsf{Win} \mid \overline{\mathsf{GoodSpan}}] \geq \frac{1}{2} \cdot \left(1 - e^{-\Omega(\lambda)}\right)$.

**Proof.** Recall that $\mathsf{FindKey}$ denotes the event in which $\mathcal{A}$ finds $(\mathsf{msk}^*, \mathsf{pp}^*, \mathsf{sk}_i^*)$ in Step 3 that satisfies the required properties. Therefore,

$$\begin{aligned}\Pr[\mathsf{Win} \mid \overline{\mathsf{GoodSpan}}] &= \Pr[\mathsf{Win} \mid \mathsf{FindKey} \cap \overline{\mathsf{GoodSpan}}] \cdot \Pr[\mathsf{FindKey} \mid \overline{\mathsf{GoodSpan}}] \\ &\quad + \Pr[\mathsf{Win} \mid \overline{\mathsf{FindKey}} \cap \overline{\mathsf{GoodSpan}}] \cdot \Pr[\overline{\mathsf{FindKey}} \mid \overline{\mathsf{GoodSpan}}] \\ &= \Pr[\mathsf{Win} \mid \mathsf{FindKey} \cap \overline{\mathsf{GoodSpan}}] \cdot \Pr[\mathsf{FindKey} \mid \overline{\mathsf{GoodSpan}}] \\ &\quad + \frac{1}{2} \cdot \Pr[\overline{\mathsf{FindKey}} \mid \overline{\mathsf{GoodSpan}}] \qquad (4.4)\end{aligned}$$

Denote by $\mathsf{FoundUseful}$ the event in which $\mathcal{A}$ finds $\mathsf{sk}_i^*$ in Step 3 and the success probability of $\mathsf{sk}_i^*$ at decrypting correctly is at least $1/2$. That is, $\mathsf{FoundUseful}$ is the event in which for each $b \in \{0,1\}$ it holds that $\Pr[\mathsf{Dec}^{\mathcal{O}}(\mathsf{pp}, \mathsf{sk}_i^*, \mathsf{Enc}^{\mathcal{O}}(\mathsf{pp}, i, b)) = b] \geq 1/2$, where the probability is taken over the internal randomness of the algorithms $\mathsf{Enc}$ and $\mathsf{Dec}$. Then, $\mathsf{FoundUseful} \subseteq \mathsf{FindKey}$, and therefore,

$$\begin{aligned}&\Pr[\mathsf{Win} \mid \mathsf{FindKey} \cap \overline{\mathsf{GoodSpan}}] \\ &\quad = \Pr[\mathsf{Win} \mid \mathsf{FoundUseful} \cap \overline{\mathsf{GoodSpan}}] \cdot \Pr[\mathsf{FoundUseful} \mid \mathsf{FindKey} \cap \overline{\mathsf{GoodSpan}}] \\ &\qquad + \Pr[\mathsf{Win} \mid \mathsf{FindKey} \cap \overline{\mathsf{FoundUseful}} \cap \overline{\mathsf{GoodSpan}}] \cdot \Pr[\overline{\mathsf{FoundUseful}} \mid \mathsf{FindKey} \cap \overline{\mathsf{GoodSpan}}] \quad (4.5)\end{aligned}$$

Now, it holds that

$$\Pr[\mathsf{Win} \mid \mathsf{FoundUseful} \cap \overline{\mathsf{GoodSpan}}] \geq \frac{1}{2} \qquad (4.6)$$

and that

$$\begin{aligned}&\Pr[\mathsf{Win} \mid \mathsf{FindKey} \cap \overline{\mathsf{FoundUseful}} \cap \overline{\mathsf{GoodSpan}}] \\ &\qquad \geq \Pr[\mathsf{Win} \mid \overline{\mathsf{Pass}} \cap \mathsf{FindKey} \cap \overline{\mathsf{FoundUseful}} \cap \overline{\mathsf{GoodSpan}}] \\ &\qquad\quad \cdot \Pr[\overline{\mathsf{Pass}} \mid \mathsf{FindKey} \cap \overline{\mathsf{FoundUseful}} \cap \overline{\mathsf{GoodSpan}}] \qquad (4.7)\end{aligned}$$

Recall that in Step 5 $\mathsf{sk}_i^*$ passes the test when decryption is correct for more than $\lambda \cdot \frac{11}{20} \cdot \left(1 - \frac{1}{20}\right) = 0.5225 \cdot \lambda$ times out of $\lambda$ times. Therefore, by Chernoff's bound,

$$\Pr[\mathsf{Pass} \mid \mathsf{FindKey} \cap \overline{\mathsf{FoundUseful}} \cap \overline{\mathsf{GoodSpan}}] \leq e^{-\frac{(0.0225)^2}{3} \cdot \frac{1}{2} \cdot \lambda} = e^{-\Omega(\lambda)}. \qquad (4.8)$$

In addition,

$$\Pr[\mathsf{Win} \mid \overline{\mathsf{Pass}} \cap \mathsf{FindKey} \cap \overline{\mathsf{FoundUseful}} \cap \overline{\mathsf{GoodSpan}}] = \frac{1}{2} \qquad (4.9)$$

Thus, combining Eq. (4.7), (4.8) and (4.9) we obtain

$$\Pr[\mathsf{Win} \mid \mathsf{FindKey} \cap \overline{\mathsf{FoundUseful}} \cap \overline{\mathsf{GoodSpan}}] \geq \frac{1}{2} \cdot \left(1 - e^{-\Omega(\lambda)}\right) \qquad (4.10)$$

and combining Eq. (4.5), (4.6) and (4.10) we obtain

$$\Pr[\mathsf{Win} \mid \mathsf{FindKey} \cap \overline{\mathsf{GoodSpan}}] \geq \frac{1}{2} \cdot \left(1 - e^{-\Omega(\lambda)}\right). \qquad (4.11)$$

Finally, combining Eq. (4.4) and (4.11) we obtain

$$\Pr[\mathsf{Win} \mid \overline{\mathsf{GoodSpan}}] \geq \frac{1}{2} \cdot \left(1 - e^{-\Omega(\lambda)}\right).$$

■

**Claim 4.9.** $\Pr\left[\mathsf{Expt}_{\mathcal{IBE},\mathcal{A}}(\lambda) = 1\right] \geq \frac{1}{2} + \frac{1}{40(n_{\mathsf{pp}}+1)} - e^{-\Omega(\lambda)}$

**Proof.** From Claim 4.7 and Claim 4.8 we obtain

$$\begin{aligned}
\Pr[\mathsf{Win}] &= \Pr[\mathsf{Win} \mid \mathsf{GoodSpan}] \cdot \Pr[\mathsf{GoodSpan}] + \Pr[\mathsf{Win} \mid \overline{\mathsf{GoodSpan}}] \cdot \Pr[\overline{\mathsf{GoodSpan}}] \\
&\geq \frac{11}{20} \cdot \left(1 - e^{-\Omega(\lambda)}\right) \cdot \Pr[\mathsf{GoodSpan}] + \frac{1}{2} \cdot \left(1 - e^{-\Omega(\lambda)}\right) \cdot (1 - \Pr[\mathsf{GoodSpan}]) \\
&= \frac{1}{2} + \left(\frac{11}{20} - \frac{1}{2}\right) \cdot \Pr[\mathsf{GoodSpan}] - e^{-\Omega(\lambda)} \\
&= \frac{1}{2} + \frac{1}{20} \cdot \Pr[\mathsf{GoodSpan}] - e^{-\Omega(\lambda)}.
\end{aligned}$$

Lemma 4.3 now implies that

$$\begin{aligned}
\Pr[\mathsf{Win}] &\geq \frac{1}{2} + \frac{1}{20} \cdot \frac{1}{2(n_{\mathsf{pp}}+1)} - e^{-\Omega(\lambda)} \\
&= \frac{1}{2} + \frac{1}{40(n_{\mathsf{pp}}+1)} - e^{-\Omega(\lambda)}
\end{aligned}$$

■

This settles the proof of Theorem 4.1. ■

## References

[ABB10] S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *Advances in Cryptology – EUROCRYPT '10*, pages 553–572, 2010.

[BB04a] D. Boneh and X. Boyen. Efficient selective-ID secure identity-based encryption without random oracles. In *Advances in Cryptology – EUROCRYPT '04*, pages 223–238, 2004.

[BB04b] D. Boneh and X. Boyen. Secure identity based encryption without random oracles. In *Advances in Cryptology – CRYPTO '04*, pages 443–459, 2004.

[BF01] D. Boneh and M. K. Franklin. Identity-based encryption from the Weil pairing. In *Advances in Cryptology – CRYPTO '01*, pages 213–229, 2001.

[BPR+08] D. Boneh, P. A. Papakonstantinou, C. Rackoff, Y. Vahlis, and B. Waters. On the impossibility of basing identity based encryption on trapdoor permutations. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 283–292, 2008.

[BR93] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pages 62–73, 1993.

[CHK03] R. Canetti, S. Halevi, and J. Katz. A forward-secure public-key encryption scheme. In *Advances in Cryptology – EUROCRYPT '03*, pages 255–271, 2003.

[CHK+10] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In *Advances in Cryptology – EUROCRYPT '10*, pages 523–552, 2010.

[Coc01] C. Cocks. An identity based encryption scheme based on quadratic residues. In *Proceedings of the 8th IMA International Conference on Cryptography and Coding*, pages 360–363, 2001.

[DG17a] N. Döttling and S. Garg. From selective IBE to full IBE and selective HIBE. In *Proceedings of the 15th Theory of Cryptography Conference*, pages 372–408, 2017.

[DG17b] N. Döttling and S. Garg. Identity-based encryption from the Diffie-Hellman assumption. In *Advances in Cryptology – CRYPTO '17*, pages 537–569, 2017.

[GPV08] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the 40th Annual ACM Symposium on Theory of computing*, pages 197–206, 2008.

[JS08] T. Jager and J. Schwenk. On the equivalence of generic group models. In *Proceedings of the 2nd International Conference on Provable Security*, pages 200–209, 2008.

[Mau05] U. Maurer. Abstract models of computation in cryptography. In *Proceedings of the 10th IMA International Conference on Cryptography and Coding*, pages 1–12, 2005.

[PRV12] P. A. Papakonstantinou, C. W. Rackoff, and Y. Vahlis. How powerful are the DDH hard groups? Cryptology ePrint Archive, Report 2012/653, 2012.

[Sha84] A. Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology – CRYPTO '84*, pages 47–53, 1984.

[Sho97] V. Shoup. Lower bounds for discrete logarithms and related problems. In *Advances in Cryptology – EUROCRYPT '97*, pages 256–266, 1997.

[Wat05] B. Waters. Efficient identity-based encryption without random oracles. In *Advances in Cryptology – EUROCRYPT '05*, pages 114–127, 2005.

## A Proof of Lemma 3.2

The random variables $X_1, \ldots, X_k$ are independent and identically distributed, and therefore for every $i \in [k]$ it holds that

$$\Pr\left[X_k \nsubseteq \mathsf{span}\left(X_1 \cup \cdots \cup X_{k-1}\right)\right] = \Pr\left[X_i \nsubseteq \mathsf{span}\left(X_1 \cup \cdots \cup X_{i-1} \cup X_{i+1} \cup \cdots \cup X_k\right)\right].$$

Thus,

$$\begin{aligned}
&\Pr\left[X_k \nsubseteq \mathsf{span}\left(X_1 \cup \cdots \cup X_{k-1}\right)\right] \\
&\qquad = \frac{1}{k} \cdot \sum_{i=1}^{k} \Pr\left[X_i \nsubseteq \mathsf{span}\left(X_1 \cup \cdots \cup X_{i-1} \cup X_{i+1} \cup \cdots \cup X_k\right)\right] \\
&\qquad = \frac{1}{k} \cdot \sum_{i=1}^{k} \sum_{U_1,\ldots,U_k \subseteq V} \Pr\left[(X_1,\ldots,X_k) = (U_1,\ldots,U_k)\right] \cdot \mathbf{1}_{\{U_i \nsubseteq \mathsf{span}(U_1 \cup \cdots \cup U_{i-1} \cup U_{i+1} \cup \cdots \cup U_k)\}} \\
&\qquad = \frac{1}{k} \cdot \sum_{U_1,\ldots,U_k \subseteq V} \Pr\left[(X_1,\ldots,X_k) = (U_1,\ldots,U_k)\right] \cdot \sum_{i=1}^{k} \mathbf{1}_{\{U_i \nsubseteq \mathsf{span}(U_1 \cup \cdots \cup U_{i-1} \cup U_{i+1} \cup \cdots \cup U_k)\}},
\end{aligned}$$

where for any event $\mathcal{E}$ we denote by $\mathbf{1}_{\mathcal{E}}$ its indicator. Since the vector space $V$ is of dimension $\mathsf{dim}(V)$, then for any $U_1, \ldots, U_k \subseteq V$ there are at most $\mathsf{dim}(V)$ indices $i \in [k]$ for which $U_i \nsubseteq \mathsf{span}\left(U_1 \cup \cdots \cup U_{i-1} \cup U_{i+1} \cup \cdots \cup U_k\right)$. Therefore,

$$\begin{aligned}
\Pr\left[X_k \nsubseteq \mathsf{span}\left(X_1 \cup \cdots \cup X_{k-1}\right)\right] &\leq \frac{1}{k} \cdot \sum_{U_1,\ldots,U_k \subseteq V} \Pr\left[(X_1, \ldots, X_k) = (U_1, \ldots, U_k)\right] \cdot \mathsf{dim}(V) \\
&= \frac{\mathsf{dim}(V)}{k}.
\end{aligned}$$

■

## B Proof of Lemma 4.2

Our proof of Lemma 4.2 relies on the following lemma (note that, unlike in the statement of Lemma 3.2, here the random variables $X_1, \ldots, X_k$ are not assumed to be independent or identically distributed):

**Lemma B.1.** *Let $k \geq 1$, and let $X_1, \ldots, X_k$ be random variables over subsets of a linear vector space $V$ of dimension $\mathsf{dim}(V)$. Let $Y$ be distributed uniformly over $\{1, \ldots, k\}$ and independent of $X_1, \ldots, X_k$. Then,*

$$\Pr_{X_1,\ldots,X_k,Y}\left[X_Y \nsubseteq \mathsf{span}\left(X_1 \cup \cdots \cup X_{Y-1}\right)\right] \leq \frac{\mathsf{dim}(V)}{k}.$$

**Proof of Lemma B.1.** Observe that

$$\begin{aligned}
&\Pr_{X_1,\ldots,X_k,Y}\left[X_Y \nsubseteq \mathsf{span}\left(X_1 \cup \cdots \cup X_{Y-1}\right)\right] \\
&= \sum_{i=1}^{k} \sum_{U_1,\ldots,U_k \subseteq V} \Pr_{X_1,\ldots,X_k,Y}[Y = i \wedge (X_1, \ldots, X_k) = (U_1, \ldots, U_k)] \cdot \mathbf{1}_{\{U_i \nsubseteq \mathsf{span}(U_1 \cup \cdots \cup U_{i-1})\}} \\
&= \sum_{i=1}^{k} \sum_{U_1,\ldots,U_k \subseteq V} \Pr_{Y}[Y = i] \cdot \Pr_{X_1,\ldots,X_k}[(X_1, \ldots, X_k) = (U_1, \ldots, U_k)] \cdot \mathbf{1}_{\{U_i \nsubseteq \mathsf{span}(U_1 \cup \cdots \cup U_{i-1})\}} \quad \text{(B.1)} \\
&= \frac{1}{k} \cdot \sum_{U_1,\ldots,U_k \subseteq V} \Pr_{X_1,\ldots,X_k}[(X_1, \ldots, X_k) = (U_1, \ldots, U_k)] \cdot \left(\sum_{i=1}^{k} \mathbf{1}_{\{U_i \nsubseteq \mathsf{span}(U_1 \cup \cdots \cup U_{i-1})\}}\right) \quad \text{(B.2)} \\
&\leq \frac{1}{k} \cdot \sum_{U_1,\ldots,U_k \subseteq V} \Pr_{X_1,\ldots,X_k}[(X_1, \ldots, X_k) = (U_1, \ldots, U_k)] \cdot \mathsf{dim}(V) \quad \text{(B.3)} \\
&= \frac{\mathsf{dim}(V)}{k}
\end{aligned}$$

where Eq. (B.1) follows from the fact that $Y$ is independent of $X_1, \ldots, X_k$, Eq. (B.2) follows from the fact that $Y$ is uniformly distributed, and Eq. (B.3) follows from the fact that $V$ is of dimension $\mathsf{dim}V$. ■

Equipped with Lemma B.1, we now prove Lemma 4.2.

**Proof of Lemma 4.2.** On the one hand, Lemma B.1 implies that

$$\frac{k - \mathsf{dim}V}{k} \leq \Pr\left[X_Y \subseteq \mathsf{span}\left(X_1 \cup \cdots \cup X_{Y-1}\right)\right].$$

On the other hand,

$$\begin{aligned}
&\Pr\left[X_Y \subseteq \mathsf{span}\left(X_1 \cup \cdots \cup X_{Y-1}\right)\right] \\
&\qquad = \Pr\left[X_Y \subseteq \mathsf{span}\left(X_1 \cup \cdots \cup X_{Y-1}\right) \mid (Y, X_1, \ldots, X_k) \in \mathsf{GoodSpan}\right] \cdot \Pr[\mathsf{GoodSpan}] \\
&\qquad\quad + \Pr\left[X_Y \subseteq \mathsf{span}\left(X_1 \cup \cdots \cup X_{Y-1}\right) \mid (Y, X_1, \ldots, X_k) \in \overline{\mathsf{GoodSpan}}\right] \cdot \Pr[\overline{\mathsf{GoodSpan}}] \\
&\qquad \leq \Pr[\mathsf{GoodSpan}] + \frac{k - \mathsf{dim} V}{2k}.
\end{aligned}$$

Therefore,

$$\Pr[\mathsf{GoodSpan}] \geq \frac{k - \mathsf{dim} V}{2k}.$$

■